TECHNISCHE UNIVERSITÄT BERLIN

Fachbereich 20 (Informatik)

Otto-Suhr-Allee 18/20

D-1000 Berlin 10

Germany

OPERATIONELLE UND FUNKTIONALE SEMANTIK VON Σ-GRAPHEN MIT ANWENDUNGEN AUF LISP

Peter Padawitz

Bericht Nr. 78-23

Dieser Bericht ist im Wortlaut identisch mit:

Peter Padawitz,
Church-Rosser-Eigenschaften von Graph-Grammatiken und Anwendungen auf die Semantik von LISP,
Diplomarbeit 1978.

Vertrieb: Universitätsbibliothek der Technischen Universität
Berlin, Abt. Publikationen,
Straße des 17. Juni 135,
D-1000 Berlin 12

ABSTRACT

Previous studies of operational versus functional semantics of symbolic expressions mostly have been confined to treelike expressions and evaluation by "simplification" and substitution of recursive definitions for function symbols. In order to drop these restrictions we introduce Σ-graphs and Σ-grammars to represent expressions and evaluation rules, respectively.

Functional semantics of Σ-graphs is defined as an extension of Scott's fixed point semantics of flow diagrams. We prove that derivations via a Σ-grammar P preserve the functional semantics of Σ-graphs if the underlying "semantic algebra" satisfies the equations given by P.

To get an operational semantics of a Σ-graph G relative to a Σ-grammar P derivations of G via P must yield a unique normal form. Therefore sufficient conditions for a weak Church-Rosser property of Σ-grammars are formulated and proved for some classes of such grammars.

Applying these results to the programming language LISP we show that the evaluation rules of a LISP interpreter are compatible with the semantics of LISP and weak Church-Rosser where garbage collection is included.

INHALT

EINLEITUNG

Ausgehend von der Theorie kontextfreier Sprachen werden in den Arbeiten von Cadiou, Manna (/CM 72/), Nivat (/Niv 74/), Rosen (/Ros 73/) und Vuillemin (/Vui 74/) symbolische Ausdrükke und deren Auswertung im Prinzip durch den folgenden Kalkül beschrieben: Man unterscheidet zwischen einer Menge Bas von Symbolen für Basisfunktionen und Konstanten und einer Menge Rek von Symbolen für rekursiv definierte Funktionen. Die Auswertung eines aus Elementen von Bas und Rek zusammengesetzten Ausdrucks entspricht einer Ableitung über einer Menge Sim von SIMPLIFIKATIONSREGELN und einer Menge Sub von SUBSTITUTIONSREGELN. Sim enthält für jede Basisfunktion f und jedes Tupel a von Konstanten im Definitionsbereich von f eine Regel

$$fa \longrightarrow f(a) \ .$$

fa ist ein symbolischer Ausdruck, während f(a) den Wert von f an der Stelle a bezeichnet. Sub enthält für alle $F \varepsilon$ Rek eine Regel $$Fx \longrightarrow \tau[F,x] \ ,$$
wobei x ein Tupel von Variablen ist, die durch beliebige Ausdrücke ersetzt werden können, und $\tau[F,x]$ einen Ausdruck bezeichnet, in dem (nicht notwendig) F oder x vorkommt.

Jeder aus Elementen von Bas und Rek zusammengesetzte Ausdruck läßt sich als endlicher Baum darstellen. Daher kann man Sim und Sub als eine GRAPH-GRAMMATIK $\mathcal{S}$ repräsentieren. Da die Produktionen von $\mathcal{S}$ PARALLEL UNABHÄNGIG sind, ist $\mathcal{S}$ ein CHURCH-ROSSER-SYSTEM (vgl. /Ros 73/,Lemma 7.4.4 und /ER 76/, Kor. 7.12).

Aus dem Ausdruck τ bilden wir $\bar{\tau}$, indem wir jedes in τ auftretende Funktionssymbol $F \varepsilon$ Rek durch den KLEINSTEN FIXPUNKT der zu F gehörigen Substitutionsregel ersetzen. $\mathcal{S}$ ist KORREKT, d.h. jede endliche Ableitungssequenz über $\mathcal{S}$, die mit τ beginnt, endet mit einer Konstanten, die dem Ergebnis der bottom-up-Auswertung von $\bar{\tau}$ entspricht (vgl. /Ros 73/,Th.8.4). Diese Eigenschaft wird als Äquivalenz von operationeller und funktionaler Semantik bezeichnet.

Stellt man jede Substitutionsregel $Fx \longrightarrow \tau[F,x]$ so als Graphproduktion dar, daß alle "occurrences" von x in

$\tau[F,x]$ identifiziert sind, dann wird bei Ableitungen über Sub die Vervielfachung von Teilausdrücken verhindert (vgl. /Vui 74/, Kap.3). Deshalb haben Ehrig und Rosen die strikte Baumdarstellung symbolischer Ausdrücke zugunsten azyklischer Graphen aufgegeben (vgl. /ER 76/,Kap.7). Schließlich konnte in /ER 77/ gezeigt werden, daß $\mathcal{S}$ zusammen mit INDIRECTION und GARBAGE COLLECTION wieder ein Church-Rosser-System ist.

In zweierlei Hinsicht werden in der vorliegenden Arbeit die bisherigen Konzepte erweitert. Erstens betrachten wir symbolische Ausdrücke beliebiger Struktur (Σ-GRAPHEN). Daher lassen sich durch die Erzeugung von Zyklen LABEL-Operatoren auswerten. Zweitens ist die Menge der Auswertungsregeln nicht auf $\mathcal{S}$ beschränkt. Da auf der linken Seite einer Produktion ein beliebiger Σ-Graph (i.a. mit Variablen) stehen kann, ist LAZY EVALUATION (vgl. /HM 76/ und /Bau 78/) möglich: Mit geeigneten Regeln können all die Teile aus einem auszuwertenden Ausdruck τ entfernt werden, die für den "Wert" von τ irrelevant sind. Dem Prinzip des lazy evaluation dienen auch die REDUKTIONSSYSTEME von Guttag, Horowitz und Musser (vgl. /GHM 76/), mit denen ein Programm auf die das Ergebnis einer Ausführung tatsächlich bestimmenden Teile reduziert wird.

Wir stellen symbolische Ausdrücke als SPEICHERZUSTÄNDE dar. In Kapitel I definieren wir ihre Repräsentationen als Graphen und (unendliche) Bäume sowie ihre (funktionale/denotationelle) Semantik bzgl. einer Interpretation der Funktionssymbole. TRANSFORMATIONSREGELN und ihre Darstellung als Graphproduktionen führen in Kapitel II zur Definition von Σ-GRAMMATIKEN. In Kapitel III zeigen wir, daß sich die Semantikverträglichkeit einer Σ-Grammatik $\mathcal{G}$ auf Ableitungen über $\mathcal{G}$ überträgt.

Ausgehend vom Einbettungstheorem für Ableitungssequenzen (vgl. /EK 76/) formulieren wir in Kapitel IV hinreichende Bedingungen für eine schwache Church-Rosser-Eigenschaft von Σ-Grammatiken und weisen diese Bedingungen für einige Klassen von Σ-Grammatiken nach. Dabei beschränken wir uns auf eine anschauliche Argumentation, weil uns noch die Techniken fehlen, mit denen sich der hohe Aufwand für einen formalen Beweis auf ein überschaubares Maß reduzieren ließe. Ferner geben wir ein "lokales" Kriterium für die parallele Unabhängigkeit der Regeln einer Σ-Grammatik an und erweitern gegenüber /ER 77/ die

Klasse der mit garbage collection vertauschbaren Regeln.

In Kapitel V werden LISP-Ausdrücke als Σ-Graphen und die in /HM 76/ angegebenen Regeln zur Auswertung von LISP-Ausdrükken als Σ-Grammatik $\mathcal{G}$ formuliert. Mit Hilfe der Ergebnisse aus Kapitel III bzw. IV läßt sich die Verträglichkeit von $\mathcal{G}$ mit der Semantik von LISP und die schwache Church-Rosser-Eigenschaft von $\mathcal{G}$ einschließlich garbage collection zeigen.

Gegenüber früheren Formulierungen konnten einige Begriffsbildungen vereinfacht und dadurch einsichtiger und leichter handhabbar gemacht werden. Ich danke Herrn Prof. Dr. H. Ehrig für wertvolle Anregungen in dieser und anderer Richtung.

NOTATIONEN

ω ist die Menge der natürlichen Zahlen.

Sei $x = (x_1, \ldots, x_n)$. Dann ist $(x)_i = x_i$ für alle $1 \leq i \leq n$.

Für die Komposition zweier Funktionen $f: A \longrightarrow B$ und $g: B \longrightarrow C$ schreiben wir $g \circ f$ oder gf. Für alle $x \varepsilon A$ ist $f(x)$ und fx der Wert von f an der Stelle x.

Spezielle Funktionen sind die Inklusion inc und die Identität id.

Den Definitionsbereich einer partiellen Funktion h bezeichnen wir mit Def(h).

Sind A und B Mengen, dann schreiben wir $|A|$ für die Anzahl der Elemente von A und $A \dot{\cup} B$ für die disjunkte Vereinigung von A und B.

Die PUNKTE eines Graphen sind seine Knoten und Kanten.

Die von einem Knoten x ausgehenden Kanten heißen OUTARCS von x. OUTDEGREE(x) ist die Anzahl der outarcs von x. Entsprechend sind INARCS und INDEGREE definiert.

Seien x und y zwei Knoten eines Graphen. Ein gerichteter Kantenzug von x nach y wird auch WEG genannt. Existiert ein Weg von x nach y, dann ist y von x aus ERREICHBAR.

Die Bedeutung weiterer Notationen findet man über den Index der Definitionen.

Kapitel I

SPEICHERZUSTÄNDE UND IHRE REPRÄSENTATIONEN

Zur Definition von Speicherzuständen und deren Semantik machen wir Gebrauch von der Theorie mehrsortiger oder heterogener Algebren (/BL 70/), die in den Arbeiten von Goguen, Thatcher u.a. die Basis der algebraischen Semantik von Programmiersprachen (/GTWW 77/) und der algebraischen Spezifikation von Datenstrukturen (/GTW 76/) bildet.

Sei Σ ein Operatorenbereich. Ein aus Elementen von Σ aufgebautes "iteratives Programm" entspricht einem endlichen Term über Σ . Demgegenüber läßt sich ein "rekursives Programm" durch einen unendlichen Term über Σ repräsentieren, der die "Expansion" des Programms wiedergibt. Eine Menge CT_{Σ} unendlicher Terme über einem geeigneten Σ erhält man auch durch die "Entfaltungen" von Flußdiagrammen (vgl. /Scott 71/ und /GTWW 77/, Kap.5.2). Als Verallgemeinerungen von Flußdiagrammen auf beliebige Operatorenbereiche führen wir Σ-Graphen ein. Algebraisch sind Σ-Graphen als parametrisierte Gleichungssysteme im Sinne von /GTWW 77/ darstellbar, die in CT_{Σ} eindeutige Lösungen haben.

Damit gibt es neben der Graphrepräsentation eines Speicherzustandes eine eindeutige Baumrepräsentation. Mit einer Abschwächung des Fixpunktsatzes von Tarski (/Tar 55/) erhält man Lösungen parametrisierter Gleichungssysteme in allen ω-stetigen Σ-Algebren (vgl. /GTWW 77/,Kap.4). Bezogen auf Σ-Graphen bezeichnen wir diese Lösungen als deren funktionale Semantiken. Wir weisen nach, daß Graphmorphismen semantikverträglich sind.

1 <u>Definition</u> (vgl. /GTWW 77/, Kap.2) :

Sei S eine Menge. Ein S-SORTIGER OPERATORENBEREICH Σ ist eine Familie von Mengen $\Sigma_{w,s}$ für alle $w \varepsilon S^*$ und $s \varepsilon S$ (S^* ist das freie Monoid über S).

$\sigma \varepsilon \Sigma_{w,s}$ heißt OPERATIONSSYMBOL mit der ARITÄT w, der SORTE s und dem RANG Lg(w) (Lg(w) ist die Länge von w). Wir schreiben auch $\sigma : w \longrightarrow s$ für $\sigma \varepsilon \Sigma_{w,s}$.

2 <u>Definition</u> :

Sei Σ ein S-sortiger Operatorenbereich.
Ein S-SORTIGER SPEICHER Sp = (X, Y, Typ) besteht aus einer Menge X von INNEREN KNOTEN, einer Menge Y von PARAMETERN mit $X \cap Y = \emptyset$ und einer Funktion

$$\text{Typ} : X \cup Y \longrightarrow S.$$

Ein Tripel (D, P, μ) heißt SPEICHERZUSTAND von Sp bzgl. Σ, wenn $D \subset X$, $P \subset Y$ und $\mu : D \longrightarrow \Sigma \times (D \cup P)^*$ eine Abbildung ist derart, daß für alle $x \varepsilon D$ und $\mu x = (\sigma, x_1, \ldots, x_n)$ gilt :

1) Sorte(σ) = Typ(x)

2) Arität(σ) = (Typ(x_1),...,Typ(x_n))

3) Rang(σ) = n

Wir schreiben μ anstatt (D, P, μ), wenn nicht explizit auf D und P Bezug genommen wird.

3 <u>Generalvoraussetzung</u>

Sei Σ ein S-sortiger Operatorenbereich und Sp = (X, Y, Typ) ein S-sortiger Speicher.

4 <u>Definition</u> :

Sei C eine Menge. Ein C-MARKIERTER GERICHTETER GRAPH $G = (N,E,s,t,m)$ setzt sich zusammen aus der KNOTENMENGE N, der KANTENMENGE E, der QUELL- und ZIELFUNKTION s bzw. t von E nach N sowie der MARKIERUNGSFUNKTION m von $N \dot{\cup} E$ nach C.

Ein Paar $g = (g_N, g_E)$ zwischen den Knoten- bzw. Kantenmengen zweier Graphen, das mit den jeweiligen Quell- und Zielfunktionen verträglich ist, heißt GRAPHMORPHISMUS. Der Einfachheit halber schreiben wir g auch für g_N oder g_E.

C-markierte Graphen und markierungsverträgliche Graphmorphismen bilden die Kategorie C-GRAPH.

5 <u>Definition</u> :

Sei $C_{fix} = \Sigma \cup \omega$("feste Markierungen"),

$C_{var} = S \times \omega$("variable Markierungen"),

$C = C_{fix} \cup C_{var}$.

Ein C-markierter gerichteter Graph $G = (N,E,s,t,m)$ heißt Σ-GRAPH, falls gilt :

1) $x \in N \implies mx \in \Sigma \cup C_{var}$

2) $x \in N,\ mx \in \Sigma \implies \text{outdegree}(x) = \text{Rang}(mx),$

$ms^{-1}x = \{1,\ldots,\text{Rang}(mx)\}$

3) $x \in N,\ mx \in C_{var} \implies \text{outdegree}(x) = 0$

Ist S einelementig, dann entsprechen Σ-Graphen den expression-Graphen in /ER 76/, Kap.7.

Sind G und H Σ-Graphen mit Markierungsfunktionen m_G bzw. m_H, dann heißt ein Graphmorphismus $g : G \longrightarrow H$ Σ-GRAPHMORPHISMUS, wenn gilt :

4) $m_G x \in C_{fix} \implies m_G x = m_H g x$

5) $m_G x \in C_{var} \implies m_G x = m_H g x$ oder

$m_H g x \in \Sigma$ und $(m_G x)_1 = \text{Sorte}(m_H g x)$

Wir erhalten die folgende bijektive Beziehung zwischen Speicherzuständen und Σ-Graphen :

6 Proposition

Sei (D,P,μ) ein Speicherzustand. Dann ist $Gr(\mu) = (N,E,s,t,m)$ ein Σ-Graph, wobei

1) $N = D \cup P$

2) $x \in D \Longrightarrow mx = (\mu x)_1$

3) $P = \{x_1, \ldots, x_n\} \Longrightarrow mx_i = (Typ(x_i), i)$
für alle $1 \leqslant i \leqslant n$

4) $E = \{(x,i,y) \in N \times \omega \times N \mid (\mu x)_{i+1} = y\}$

5) $(x,i,y) \in E \Longrightarrow s(x,i,y) = x,\ t(x,i,y) = y,$
$m(x,i,y) = i$

Wir bezeichnen $Gr(\mu)$ als GRAPHREPRÄSENTATION von μ.

Sei $G = (N,E,s,t,m)$ ein Σ-Graph. Dann ist $Sz(G) = (D,P,\mu)$ ein Speicherzustand, wobei

6) $D = \{x \in N \mid mx \in \Sigma\}$

7) $P = N - D$

8) $x \in D \Longrightarrow (\mu x)_1 = mx$

9) $e \in E,\ me = i \Longrightarrow (\mu se)_{i+1} = te$

Sz(G) heißt ZUSTANDSREPRÄSENTATION von G.

Man zeigt leicht, daß die Operatoren Gr und Sz invers zueinander sind.

7 Proposition

Seien G und H Σ-Graphen mit den Markierungsfunktionen m_G bzw. m_H, $g: G \longrightarrow H$ Σ-Graphmorphismus, $Sz(G) = (D,P,\mu)$ und $Sz(H) = (D',P',\eta)$. Dann gilt

$$gD \subset D' .$$

Beweis:

$x \in D \Longrightarrow m_G x \in \Sigma \Longrightarrow m_H gx = m_G x \in \Sigma \Longrightarrow gx \in D'$.

Im folgenden entwickeln wir die Darstellung von Speicherzuständen als parametrisierte Gleichungssysteme und definieren ihre funktionale Semantik.

8 <u>Definition</u> (vgl. /GTWW 77/, Kap.2) :

Sei $D \subset X$. Die Menge $T_\Sigma(D)$ der Σ-TERME über D ist induktiv definiert durch

1) $D \cup \Sigma_{\lambda,s} \subset T_\Sigma(D)_s$ für alle $s \in S$

2) $\sigma: s_1 \ldots s_n \longrightarrow s,\ t_i \in T_\Sigma(D)_{s_i}$ für alle $1 \leq i \leq n$
$\Longrightarrow \sigma(t_1, \ldots, t_n) \in T_\Sigma(D)_s$

3) $T_\Sigma(D) = \bigcup_{s \in S} T_\Sigma(D)_s$

9 <u>Definition</u> (vgl. /GTWW 77/, Kap.4) :

Eine Σ-ALGEBRA A besteht aus einer Trägermenge A_s für jedes $s \in S$ und einer Operation $\sigma_A: A_{s_1} \times \ldots \times A_{s_n} \longrightarrow A_s$ für jedes $\sigma: s_1 \ldots s_n \longrightarrow s$. Für $\sigma: \lambda \longrightarrow s$ gilt $\sigma_A \in A_s$. σ_A heißt dann Konstante.

A ist ω-STETIG, wenn für alle $s \in S$ eine Halbordnung $\leq_s$ auf A_s mit Suprema von ω-Ketten und kleinstem Element $\perp_s$ existiert und alle Operationen diese Suprema bewahren. Ist $\{a_i\}_{i \in \omega}$ eine ω-Kette in A, dann schreiben wir $\bigsqcup_{i \in \omega} a_i$ für ihr Supremum. Außerdem sei $\leq := \bigcup_{s \in S} \leq_s$.

Sei $D \subset X \cup Y$. Dann bezeichnen wir mit A^D die Menge der TYPVERTRÄGLICHEN Funktionen $f: D \longrightarrow A$, d.h.

$$fx \in A_{Typ(x)} \quad \text{für alle } x \in D.$$

Wie aus Def.8 ersichtlich ist, kann man für jede Teilmenge D von X $T_\Sigma(D)$ als Σ-Algebra auffassen.

10 <u>Definition</u> (vgl. /GTWW 77/, 5.1) :

Sei $D \subset X$. Eine typverträgliche Funktion $E:D \longrightarrow T_{\Sigma}(D)$ heißt GLEICHUNGSSYSTEM in T_{Σ}.

11 <u>Proposition</u>

Sei $E:D \longrightarrow T_{\Sigma}(D)$ ein Gleichungssystem in T_{Σ}, A eine ω-stetige Σ-Algebra und $b \in A^D$. Da $T_{\Sigma}(D)$ freie Σ-Algebra über D ist, existiert eine eindeutige Fortset- $b^*:T_{\Sigma}(D) \longrightarrow A$ von b. Sei $E_A:A^D \longrightarrow A^D$ definiert durch

$$E_A(b) = b^* \circ E \;.$$

E_A ist ω-stetig und hat deshalb nach dem Fixpunktsatz von Tarski (/Tar 55/) den kleinsten Fixpunkt

$$|E_A| = \bigsqcup_{n \in \omega} E_A^n(\perp) \;.$$

Jeder Fixpunkt von E_A heißt LÖSUNG von E in A.

12 <u>Definition</u> (vgl. /GTWW 77/, 5.4) :

$\Sigma(Y) := \Sigma \cup Y$ wird durch

$$\Sigma(Y)_{w,s} = \begin{cases} \Sigma_{w,s} \cup (\mathrm{Typ}^{-1}(s) \cap Y) & \text{falls } w = \lambda \\ \Sigma_{w,s} & \text{sonst} \end{cases}$$

zum S-sortigen Operatorenbereich.

Ein Gleichungssystem $F:D \longrightarrow T_{\Sigma(Y)}(D)$ heißt PARAMETRISIERT mit Parametern Y.

Sei $b \in A^Y$. Dann erhalten wir mit

$$A(b)_s = A_s \qquad \text{für alle } s \in S, \text{ und}$$

$$\sigma_{A(b)} = \begin{cases} b(\sigma) & \text{falls } \sigma \in Y \\ \sigma_A & \text{sonst} \end{cases}$$

die $\Sigma(Y)$-Algebra $A(b)$. Schließlich ist $\|F_A\|:A^Y \longrightarrow A^D$ definiert durch

$$\|F_A\|(b) = |F_{A(b)}| \;.$$

13 <u>Definition</u> :

Jeder Speicherzustand (D,P,μ) läßt sich durch ein parametrisiertes Gleichungssystem

$$\tilde{\mu} : D \longrightarrow T_{\Sigma(Y)}(D)$$

beschreiben, wobei gilt :

$$\mu x = (\sigma, x_1, \dots, x_n) \Longleftrightarrow \tilde{\mu} x = \sigma(x_1, \dots, x_n) .$$

Sei A eine ω-stetige Σ-Algebra. Dann ist $\| \tilde{\mu}_A \|$ die FUNKTIONALE SEMANTIK von μ bzgl. A.

14 <u>Proposition</u>

Sei $b \in A^Y$ und $\mu x = (\sigma, x_1, \dots, x_n)$. Dann gilt für $f = \| \tilde{\mu}_A \| (b) = | \tilde{\mu}_{A(b)} |$:

$$fx = \tilde{\mu}_{A(b)}(f)x = f^* \tilde{\mu} x = \sigma_A(f^* x_1, \dots, f^* x_n) =$$
$$= \sigma_A(fx_1, \dots, fx_n) .$$

Im folgenden wird gezeigt, wie die "Entfaltung" eines Speicherzustandes μ zu einem unendlichen Baum eine eindeutige Lösung von $\tilde{\mu}$ definiert.

15 <u>Definition</u> (vgl./GTWW 77/, Kap.4)

Die Menge CT_Σ der Σ-BÄUME ist definiert als die Menge der partiellen Funktionen $t : \omega^* \longrightarrow \Sigma$ derart, daß für alle $u \in \omega^*$ und $k \in \omega$ gilt :

1) $uk \in Def(t) \Longrightarrow u \in Def(t),\ k \leqslant Rang(tu)$

2) $uk \in Def(t) \Longrightarrow Sorte(t(uk)) = (Arität(tu))_k$

Durch folgende Definitionen wird CT_Σ zur ω-stetigen Σ-Algebra :

3) Für alle $s \in S$ ist $\perp_s$ die nirgends definierte Funktion von ω^* nach $\bigcup_{w \in \omega^*} \Sigma_{w,s}$ und $CT_{\Sigma,s} = \{ t \in CT_\Sigma \mid Sorte(t\lambda) = s \} \cup \{ \perp_s \}$.

4) $\sigma: s_1 \ldots s_n \longrightarrow s \in \Sigma$, $t_i \in CT_{\Sigma, s_i}$ für alle $1 \leq i \leq n$

$$\Longrightarrow \sigma_{CT}(t_1, \ldots, t_n)(\lambda) = \sigma$$

$$\sigma_{CT}(t_1, \ldots, t_n)(iu) = \begin{cases} t_i u & \text{falls } i \leq n \\ \text{undefiniert} & \text{sonst} \end{cases}$$

Nach /GTWW 77/, Prop.4.5 ist σ_{CT} ω-stetig.

5) $\leq_s := \{ (t,t') \in CT_{\Sigma,s} \mid u \in Def(t) \Rightarrow tu = t'u \}$

Wir zeigen, daß Gleichungssysteme in T_Σ eindeutige Lösungen in CT_Σ haben, wenn darin keine Gleichungen der Form $x = x'$ vorkommen.

16 Definition (vgl. /Elg 73/) :

Sei $D \subset X$. Ein Gleichungssystem $E: D \longrightarrow T_\Sigma(D)$ heißt IDEAL, wenn für alle $x \in D$ $Ex \notin D$ gilt.

17 Satz

Ideale Gleichungssysteme haben eindeutige Lösungen in CT_Σ.

Beweis:

Sei $E: D \longrightarrow T_\Sigma(D)$ ein ideales Gleichungssystem. Nach Prop.11 ist $|E_{CT}|$ eine Lösung von E in CT_Σ. Sei L eine weitere Lösung von E in CT_Σ. Wir zeigen durch Induktion über n für alle $x \in D$:

$$w \in Def(Lx),\ Lg(w) = n \quad \Longrightarrow \quad w \in Def(E_{CT}^{n+1}(\perp)x) \qquad (1)$$

Sei $x \subset D$. Da E ideal ist, gibt es $\sigma: s_1 \ldots s_m \longrightarrow s \in \Sigma$ und $t_i \in T_\Sigma(D)_{s_i}$ für alle $1 \leq i \leq m$ mit $Ex = \sigma(t_1, \ldots, t_m)$.

a) Sei $w = \lambda$. Es gilt $E_{CT}(\perp)x = \perp^* Ex =$

$= \sigma_{CT}(\perp^* t_1, \ldots, \perp^* t_m)$. Nach Def. von σ_{CT} (15.4) folgt $E_{CT}(\perp)(x)(\lambda) = \sigma$, also $w \in Def(E_{CT}(\perp)x)$.

b) Sei $kw \in Def(Lx)$, $Lg(kw) = n$. Da L Lösung von E ist, gilt $Lx = E_{CT}(L)x = L^*Ex = \sigma_{CT}(L^*t_1,\ldots,L^*t_m)$.

Nach Voraussetzung und Def. von σ_{CT} folgt

$$k \leqslant m \quad \text{und} \quad w \in Def(L^*t_k) \,. \tag{2}$$

Daraus erhält man nach Induktionsvoraussetzung und Lemma 18

$$w \in Def(E^n_{CT}(\perp)^*t_k). \tag{3}$$

Außerdem ist $E^{n+1}_{CT}(\perp)x = E_{CT}(E^n_{CT}(\perp))x = E^n_{CT}(\perp)^*Ex$ $= \sigma_{CT}(E^n_{CT}(\perp)^*t_1,\ldots,E^n_{CT}(\perp)^*t_m)$. Daraus folgt wegen (2) nach Def. von σ_{CT}

$$E^{n+1}_{CT}(\perp)(x)(kw) = E^n_{CT}(\perp)^*t_kw$$

und schließlich wegen (3)

$$kw \in Def(E^{n+1}_{CT}(\perp)x) \,,$$

so daß (1) gezeigt ist.

Aus (1) folgt nach Def. von E_{CT} $Def(Lx) \subset Def(|E_{CT}|x)$ für alle $x \in D$. Da $|E_{CT}|$ kleinste Lösung von E in CT ist, gilt $|E_{CT}|x \leqslant Lx$ für alle $x \in D$, und man erhält nach Def. von $\leqslant$ (15.5 und 9) $|E_{CT}| = L$.

18 <u>Lemma</u>

Seien $f,g \in CT^D_\Sigma$ und es gelte

$$w \in Def(fx),\ Lg(w) \leqslant n \implies w \in Def(gx)$$

für alle $x \in D$. Dann folgt

$$w \in Def(f^*t),\ Lg(w) \leqslant n \implies w \in Def(g^*t)$$

für alle $t \in T_\Sigma(D)$.

Beweis durch Induktion über den Aufbau von t :

a) Sei $t \in D$. $\implies f^*t = ft,\ g^*t = gt \implies$ Behauptung.

b) Sei t Konstante. $\implies f^*t = t = g^*t \implies$ Behauptung.

c) Sei $t = \sigma(t_1,\ldots,t_m)$. Dann ist

$$f^*t = \sigma_{CT}(f^*t_1,\ldots,f^*t_m),\quad g^*t = \sigma_{CT}(g^*t_1,\ldots,g^*t_m).$$

Daraus folgt $f^*t(\lambda) = \sigma = g^*t(\lambda)$ und für $kw \in Def(f^*t)$

$k \leqslant m$ und $f^*t(kw) = f^*t_k w$,

also $w \in Def(f^*t_k)$, nach Induktionsvoraussetzung $w \in Def(g^*t_k)$. Wegen $k \leqslant m$ gilt nach Def. von σ_{CT} $g^*t(kw) = g^*t_k w$, demnach $kw \in Def(g^*t)$.

19 Definition

Für jeden Speicherzustand μ ist $\tilde{\mu}$ (Def.13) ein ideales Gleichungssystem in T_Σ. Folglich gibt es nach Satz 17 eine eindeutige Lösung $Br(\mu)$ von $\tilde{\mu}$ in $CT_{\Sigma(Y)}$. $Br(\mu)$ heißt BAUMREPRÄSENTATION von μ.

Wir wollen jetzt die Semantikverträglichkeit von Σ-Graphmorphismen zeigen, ein Ergebnis, mit dem wir in Kap. III die "Korrektheit" von Graphtransformationen auf die Semantikverträglichkeit des verwendeten Regelsystems zurückführen können.

20 Voraussetzung (für 21-24)

Sei A eine ω-stetige Σ-Algebra, $G = (G_N, G_E, s_G, t_G, m_G)$ und $H = (H_N, H_E, s_H, t_H, m_H)$ Σ-Graphen, $g: G \longrightarrow H$ ein Σ-Graphmorphismus, $Sz(G) = (D, P, \mu)$, $Sz(H) = (D', P', \eta)$ und g^* die eindeutige Fortsetzung (der Knotenkomponente) von g auf $T_\Sigma(D \cup P)$.

21 Lemma

Für alle $x \in D$ gilt $\tilde{\eta} g x = g^* \tilde{\mu} x$.

Beweis:

Wegen $gD \subset D'$ (Prop.7) ist $\tilde{\eta}\, gx$ für $x \in D$ definiert.

Sei $x \in D$, $\mathrm{Rang}(m_G x) = n$. Dann gilt

$$(\eta gx)_1 = m_H gx = m_G x = (\mu x)_1 \,, \tag{1}$$

und für alle $1 \leqslant i \leqslant n$ existiert $e \in G_E$ mit $m_G e = i$ und

$$(\eta gx)_{i+1} = (\eta g s_G e)_{i+1} = (\eta s_H ge)_{i+1} = t_H ge$$
$$= g t_G e = g(\mu s_G e)_{i+1} = g(\mu x)_{i+1} \,. \tag{2}$$

Sei $\mu x = (\sigma, x_1, \ldots, x_n)$. Aus (1) und (2) folgt
$\eta gx = (\sigma, gx_1, \ldots, gx_n)$, also $\tilde{\eta} gx = \sigma(gx_1, \ldots, gx_n)$
$= \sigma(g^* x_1, \ldots, g^* x_n) = g^* \tilde{\mu} x$.

22 Lemma

Seien $b, c \in A^Y$, $f \in A(b)^D$ und $h \in A(c)^{D'}$ (vgl. Def.12)

$$\text{mit} \qquad by = \begin{cases} hgy & \text{falls } y \in P \text{ und } gy \in D' \\ cgy & \text{falls } y \in P \text{ und } gy \in P' \\ cy & \text{sonst} \end{cases}$$

und $\qquad fx = hgx \qquad$ für alle $x \in D$.

Dann gilt für die Fortsetzungen

$$f^* : T_{\Sigma(Y)}(D) \longrightarrow A(b) \text{ und } h^* : T_{\Sigma(Y)}(D') \longrightarrow A(c)$$

von f bzw. h :

$$f^* \tilde{\mu} x = h^* \tilde{\eta} gx \qquad \text{für alle } x \in D \,. \tag{1}$$

Beweis:

Nach Lemma 21 ist (1) gleichbedeutend mit

$$f^* \tilde{\mu} x = h^* g^* \tilde{\mu} x \qquad \text{für alle } x \in D \,.$$

Wegen $\mu x \in \Sigma \times (D \cup P)^*$ gilt $\tilde{\mu} x \in T_{\Sigma(P)}(D)$. Daher bleibt zu zeigen :

$$f^* t = h^* g^* t \qquad \text{für alle } t \in T_{\Sigma(P)}(D) \,. \tag{2}$$

Wir beweisen (2) durch Induktion über den Aufbau von t :

a) t Konstante in $\Sigma \Longrightarrow f^* t = t = h^* t = h^* g^* t$

b) $t \in P,\ gt \in D' \Longrightarrow f^* t = bt = hgt = h^* gt = h^* g^* t$

c) $t \in P,\ gt \in P' \Longrightarrow f^* t = bt = cgt = h^* gt = h^* g^* t$

d) $t \in D \Longrightarrow gt \in D' \Longrightarrow f^* t = ft = hgt = h^* gt = h^* g^* t$

e) $t = \sigma(t_1,\ldots,t_n)$. Nach Induktionsvoraussetzung gilt

$$f^*t = \sigma_A(f^*t_1,\ldots,f^*t_n) = \sigma_A(h^*g^*t_1,\ldots,h^*g^*t_n)$$
$$= h^*\sigma(g^*t_1,\ldots,g^*t_n) = h^*g^*t \; .$$

23 Lemma

Sei $c \in A^Y$ und für alle $n \in \omega$ sei $b_n \in A^Y$ definiert

durch
$$b_n y = \begin{cases} \tilde{\eta}^n_{A(c)}(\perp)gy & \text{falls } y \in P \text{ und } gy \in D' \\ cgy & \text{falls } y \in P \text{ und } gy \in P' \\ cy & \text{sonst} \end{cases}$$

Dann gilt für alle $x \in D$ und $n \in \omega$

$$\tilde{\eta}^n_{A(c)}(\perp)gx \leqslant \tilde{\mu}^n_{A(b_n)}(\perp)x \; .$$

Beweis durch Induktion über n :

$n = 0$ ist klar.

Wegen $b_n \leqslant b_{n+1}$ gilt

$$\tilde{\mu}^n_{A(b_n)}(\perp) \leqslant \tilde{\mu}^n_{A(b_{n+1})}(\perp) \; . \qquad (1)$$

Sei $h = \tilde{\eta}^n_{A(c)}(\perp)$ und $f \in A(b_n)^D$ definiert durch

$$fx = hgx \qquad \text{für alle } x \in D \; .$$

Mit $b = b_n$ erfüllen f und h die Voraussetzungen von Lemma 22. Nach Induktionsvoraussetzung gilt

$$f \leqslant \tilde{\mu}^n_{A(b_n)}(\perp) \; . \qquad (2)$$

Nach Lemma 22 folgt für alle $x \in D$:

$$\tilde{\mu}^{n+1}_{A(b_{n+1})}(\perp)x = \tilde{\mu}_{A(b_{n+1})}(\tilde{\mu}^n_{A(b_{n+1})}(\perp))x$$
$$\geqslant \tilde{\mu}_{A(b_{n+1})}(\tilde{\mu}^n_{A(b_n)}(\perp))x \qquad \text{wegen (1)}$$
$$\geqslant \tilde{\mu}_{A(b_{n+1})}(f)x \qquad \text{wegen (2)}$$
$$= f^*\tilde{\mu}x = h^*\tilde{\eta}gx = \tilde{\eta}_{A(c)}(h)gx = \tilde{\eta}^{n+1}_{A(c)}(\perp)gx \; .$$

Die Beweistechnik des vorangegangenen Lemmas wird auch im folgenden Satz, dem Nachweis der Semantikverträglichkeit von g, verwendet. Die Aussage von Lemma 23 benötigen wir in Kap. III.

24 Satz

Seien $b, c \in A^Y$ mit

$$by = \begin{cases} |\tilde{\eta}_{A(c)}|\, gy & \text{falls } y \in P \text{ und } gy \in D' \\ cgy & \text{falls } y \in P \text{ und } gy \in P' \\ cy & \text{sonst} \end{cases}$$

Dann gilt für alle $x \in D$

$$|\tilde{\mu}_{A(b)}|\, x = |\tilde{\eta}_{A(c)}|\, gx \,.$$

Beweis:

Wir zeigen zunächst für alle $n \in \omega$ und $x \in D$

$$\tilde{\mu}^n_{A(b)}(\perp)x \leqslant |\tilde{\eta}_{A(c)}|\, gx \qquad (1)$$

durch Induktion über n.

$n = 0$ ist klar.

Sei $h = |\tilde{\eta}_{A(c)}|$ und $f \in A(b)^D$ definiert durch

$$fx = hgx \qquad \text{für alle } x \in D \,.$$

f und h genügen den Voraussetzungen von Lemma 22.

Nach Induktionsvoraussetzung gilt

$$\tilde{\mu}^n_{A(b)}(\perp) \leqslant f \,. \qquad (2)$$

Nach Lemma 22 folgt für alle $x \in D$:

$$\tilde{\mu}^{n+1}_{A(b)}(\perp)x = \tilde{\mu}_{A(b)}(\tilde{\mu}^n_{A(b)}(\perp))x$$

$$\leqslant \tilde{\mu}_{A(b)}(f)x \qquad \text{wegen (2)}$$

$$= f^*\tilde{\mu}\, x = h^*\tilde{\eta}\, gx = \tilde{\eta}_{A(c)}(h)gx$$

$$= |\tilde{\eta}_{A(c)}|\, gx \,. \qquad (h \text{ ist Fixpunkt von } \tilde{\eta}_{A(c)})$$

Also gilt (1).

Zweitens zeigen wir für alle $n \in \omega$ und $x \in D$

$$\tilde{\eta}^n_{A(c)}(\perp)gx \leqslant \tilde{\mu}^n_{A(b)}(\perp)x \qquad (3)$$

durch Induktion über n.

$n = 0$ ist klar.

Sei $h = \tilde{\eta}^{n}_{A(c)}(\perp)$ und $f \varepsilon A(b)^D$ definiert durch

$$fx = hgx \qquad \text{für alle } x \varepsilon D .$$

f und h genügen wieder den Voraussetzungen von Lemma 22, und nach Induktionsvoraussetzung gilt jetzt

$$f \leqslant \tilde{\mu}^{n}_{A(b)}(\perp) . \tag{4}$$

Nach Lemma 22 folgt für alle $x \varepsilon D$:

$$\tilde{\mu}^{n+1}_{A(b)}(\perp)x = \tilde{\mu}_{A(b)}(\tilde{\mu}^{n}_{A(b)}(\perp))x$$

$$\geqslant \tilde{\mu}_{A(b)}(f)x \qquad \text{wegen (4)}$$

$$= f^{*}\tilde{\mu}x = h^{*}\tilde{\eta}gx = \tilde{\eta}_{A(c)}(h)gx = \tilde{\eta}^{n+1}_{A(c)}(\perp)gx .$$

Damit gilt auch (3).

Aus (1) folgt

$$|\tilde{\mu}_{A(b)}|\, x = \bigsqcup_{n\varepsilon\omega} \tilde{\mu}^{n}_{A(b)}(\perp)x \leqslant |\tilde{\eta}_{A(c)}|\, gx \tag{5}$$

und aus (3)

$$|\tilde{\eta}_{A(c)}|\, gx \leqslant |\tilde{\mu}_{A(b)}|\, x \tag{6}$$

für alle $x \varepsilon D$.

(5) und (6) ergeben die Behauptung des Satzes.

Kapitel II

ZUSTANDSTRANSFORMATIONEN

Wir wollen Transformationen von Speicherzuständen durch Ableitungen über einem System von Ersetzungsregeln beschreiben. Aufgrund der Bijektion zwischen Speicherzuständen und Σ-Graphen (vgl. I.6) lassen sich solche Ersetzungsregeln als Graphproduktionen formulieren.

Ausgehend von einer Transformationsregel, d.i. ein Paar (μ, η) von Speicherzuständen, konstruieren wir einen (Klebe-) Graphen K so, daß

$$(Gr(\mu) \xleftarrow{inc} K \xrightarrow{inc} Gr(\eta))$$

eine Graphproduktion im Sinne der algebraischen Theorie von Graph-Grammatiken (/EK 75 a/,/Ehr 78/ u.a.) wird. Graphproduktionen bzw. direkte Graphableitungen kann man als Verallgemeinerungen von rule schemas bzw. rule instances bei Baumersetzungssystemen (/Ros 73/) auffassen: Während rule schemas nur injektiv in rule instances eingebettet werden können, muß es von Graphproduktionen nach direkten Graphableitungen nur eine homomorphe Zuordnung geben.

In /ER 76/ wurde das Konzept des recoloring eingeführt, um bei der Anwendung von Graphproduktionen unterschiedliche Markierungen von "Parameter"knoten zu erlauben. Durch den Operatorenbereich Σ und die Sortenmenge S ist die Menge R_{Σ} der recoloring-Funktionen für Σ-Graphen festgelegt.

Die den Transformationsregeln (μ, η) entsprechenden Graphproduktionen bezeichnen wir als Σ-Produktionen. Σ-Produktionen sind fast (vgl. /ER 77/,/Ehr 78/), so daß die Anwendbarkeit einer Σ-Produktion nur von den Teilen des abzuleitenden Graphen abhängt, die zu ihrem Ansatz gehören. Die Menge der Σ-Graphen ist abgeschlossen bzgl. Ableitungen über Σ-Produktionen, wenn eine schwache Anwendbarkeitsbedingung erfüllt ist.

1 Definition

Ein Paar $t = ((D,P,\mu),(D',P',\eta))$ von Speicherzuständen heißt TRANSFORMATIONSREGEL, falls gilt :

1) $D \subset D'$
2) $P = P'$
3) Es existiert ein $x_t \in D$ so, daß für alle $x \in D$ mit $x \neq x_t$ $\quad \mu x = \eta x \quad$ gilt.

2 Definition

Sei C eine Menge von Markierungen (vgl. I.4). Eine GRAPHPRODUKTION

$$p = (B_1 \xleftarrow{b_1} K_1, K_2 \xrightarrow{b_2} B_2)$$

besteht aus den C-markierten Graphen

$$B_i = (B_{i,N}, B_{i,E}, s_i, t_i, m_i) ,$$
$$K_i = (K_N, K_E, s_K, t_K, m_{K,i})$$

für $i \in \{1,2\}$ und den markierungsverträglichen Graphmorphismen b_1 und b_2. B_1 und B_2 heißen LINKE bzw. RECHTE SEITE und $K = (K_N, K_E, s_K, t_K)$ KLEBEGRAPH von p. Wir schreiben kurz:

$$p = (B_1 \xleftarrow{b_1} K \xrightarrow{b_2} B_2) ,$$

weil $m_{K,i}$ wegen der Markierungsverträglichkeit von b_i durch m_i eindeutig bestimmt ist.

3 Definition

Sei $\quad p = (B_1 \xleftarrow{b_1} K \xrightarrow{b_2} B_2) \quad$ eine Graphproduktion, $G = (G_N, G_E, s_G, t_G, m_G)$ und $H = (H_N, H_E, s_H, t_H, m_H)$ C-markierte Graphen. Dann wird durch zwei Pushouts in C-GRAPH

$$\begin{tikzcd} B_1 \arrow[d, "g"'] & K_1 \arrow[l, "b_1"'] \arrow[d, "d"] & K_2 \arrow[r, "b_2"] \arrow[d, "d"'] & B_2 \arrow[d, "h"] \\ G & D_1 \arrow[l, "c_1"] & D_2 \arrow[r, "c_2"'] & H \end{tikzcd} \qquad (1)$$

mit $D_i = (D_N, D_E, s_D, t_D, m_{D,i})$ für $i \in \{1,2\}$ eine DIREKTE

ABLEITUNG von G nach H über p definiert, geschrieben

$$G \overset{p}{\Longrightarrow} H ,$$

falls für alle $x \in D\text{-}dK$ $m_{D,1}x = m_{D,2}x$ gilt, wobei $D = (D_N, D_E, s_D, t_D)$. D heißt RESTGRAPH von G und gB_1 ANSATZ von p in G.

Eine Menge P von Graphproduktionen heißt GRAPH-GRAMMATIK. Eine ABLEITUNG a von G nach H über P ist eine endliche Folge

$$\{ G_i \overset{p_i}{\Longrightarrow} G_{i+1} \}_{1 \leqslant i < n}$$

direkter Ableitungen mit $G_1 = G$, $G_n = H$ und $\{p_1, \ldots, p_n\} \subset P$. a heißt NATÜRLICH, wenn der Graphmorphismus c_1 (vgl.Fig.(1)) in allen Komponenten von a injektiv ist.

4 Definition (vgl. I.5)

Seien C_{fix} und C_{var} disjunkte Mengen und $C = C_{fix} \cup C_{var}$. Eine Funktion $r{:}C \longrightarrow C$ heißt RECOLORING, wenn für alle $c \in C_{fix}$ $rc = c$ gilt.

Eine Menge R von recolorings heißt VOLLSTÄNDIG, wenn für jede Teilmenge $\{r_c\}_{c \in C}$ von R auch die Funktion $r{:}C \longrightarrow C$ mit $rc = r_c c$ zu R gehört.

5 Proposition

Sei $G = (G_N, G_E, s_G, t_G, m_G)$ ein C-markierter Graph und $p = (B_1 \xleftarrow{b_1} K \xrightarrow{b_2} B_2)$ eine Graphproduktion. Dann ist auch $G_r = (G_N, G_E, s_G, t_G, rm_G)$ ein C-markierter Graph und $rp = (B_{1,r} \xleftarrow{b_1} K \xrightarrow{b_2} B_{2,r})$ eine Graphproduktion für alle recolorings $r{:}C \longrightarrow C$.

6 Proposition

Sei C wie in Def. I.5 und R_Σ die Menge aller recolorings $r:C \longrightarrow C$ mit
$$r(s,i) \in \mathrm{Sorte}^{-1}(s) \cup \{(s,i)\}$$
für alle $(s,i) \in C_{var}$. R_Σ ist vollständig.

Beweis:
Auf $C \times C$ sei das folgende Prädikat P definiert :
$$P(a,b) = \begin{cases} b \in \mathrm{Sorte}^{-1}(a_1) \cup \{a\} & \text{falls } a \in C_{var} \\ a = b & \text{sonst} \end{cases}$$
Dann gilt
$$r \in R_\Sigma \iff P(c,rc) \text{ für alle } c \in C\,. \qquad (1)$$
Daraus folgt für $\{r_c\}_{c \in C} \subset R$
$$P(c,r_c c) \quad \text{für alle } c \in C\,.$$
Also ist für $r:C \longrightarrow C$ mit $rc = r_c c$ $P(c,rc)$ wahr, so daß wegen (1) auch r zu R_Σ gehört.

7 Definition (vgl. /ER 76/, Kap.5)

Sei R eine Menge von recolorings. Eine Graphproduktion p heißt PROPER bzgl. R, falls gilt :

1) $x,y \in B_1,\ m_1x = m_1y \in C_{var} \implies x = y$
   (Variable Markierungen in B_1 sind eindeutig)
2) $x \in B_i,\ m_ix \in C_{var} \implies x \in b_iK \qquad (i = 1,2)$
   (Variabel markierte Punkte von B_i sind Klebepunkte)
3) $x \in K,\ m_1b_1x \in C_{var}$ oder $m_2b_2x \in C_{var}$
   $$\implies m_1b_1x = m_2b_2x$$
   (Variable Markierungen von Klebepunkten setzen sich durch p fort)
4) $x \in K,\ m_1b_1x \in C_{fix} \implies R^{-1}m_1b_1x \cap C_{var} \subset R^{-1}m_2b_2x$
   (Ist die Markierung eines Klebepunktes x in B_1 recoloring einer Variablen v, dann ist auch die Markierung von x in B_2 recoloring von v)

Bzgl. der Bezeichnungen vgl. Def.2.

8 Definition

Eine Graphproduktion $p = (B_1 \xleftarrow{b_1} K \xrightarrow{b_2} B_2)$ heißt FAST, wenn b_1 und b_2 injektiv sind und b_1 surjektiv auf Knoten ist.

9 Definition

Eine Graphproduktion p heißt Σ-PRODUKTION, falls gilt:

1) p ist proper bzgl. R_Σ

2) p ist fast

3) B_1 und B_2 sind Σ-Graphen

4) Es existiert ein Knoten x_p in K, die WURZEL von p, mit
   a) $m_1 b_1 x_p \in \Sigma$
   b) $x \in K-\{x_p\} \Longrightarrow m_1 b_1 x = m_2 b_2 x$
   c) $e \in B_{1,E} - b_1 K_E \Longrightarrow s_1 e = b_1 x_p$
   d) $e \in B_{2,E} - b_2 K_E,\ s_2 e \neq b_2 x_p \Longrightarrow s_2 e \notin b_2 K_N$

Eine Menge von Σ-Produktionen heißt Σ-GRAMMATIK.

10 Proposition

Ableitungen über Σ-Produktionen sind natürlich.

Beweis:

Die Behauptung folgt aus 9.2, da c_1 (vgl. Def.3) Pushoutfortsetzung von b_1 ist.

11 Proposition

In allen direkten Ableitungen über Σ-Produktionen sind die Markierungsfunktionen $m_{D,1}$ und $m_{D,2}$ des Restgraphen (vgl. Def.3) auf Kanten gleich.

Beweis:

Für $e \in D_E - dK_E$ gilt die Behauptung nach Def. der direkten Ableitung. Für $e = de'$ ist $m_{D,1}e = m_{K,1}e' = m_1 b_1 e'$ $= m_2 b_2 e' = m_{K,2}e' = m_{D,2}e$ nach 9.4 b.

Nach Def.3 hängt die Konstruktion direkter Ableitungen von der Existenz des Restgraphen D ab. Diese ist jedoch gesichert, wenn die folgenden Forderungen (KLEBEBEDINGUNGEN) an die Einbettung g von B_1 in G erfüllt sind. In diesem Fall ist D bis auf Isomorphie eindeutig durch die Produktion p und die Einbettung g bestimmt, wenn man sich auf natürliche Ableitungen beschränkt. Umgekehrt gelten die Klebebedingungen für g und h in allen direkten Ableitungen. Zum Beweis dieser Behauptung und zur Bedeutung der Klebebedingungen vgl. z.B. /ED 75/, Satz 3.16/17.

12 Proposition

Sei $G \overset{p}{\Longrightarrow} H$ eine direkte Ableitung (vgl. Def.3). Dann gelten für die Graphmorphismen g und h folgende KLEBEBEDINGUNGEN :

1) $gx = gy,\ x \neq y \Longrightarrow x,y \in b_1K$

$hx = hy,\ x \neq y \Longrightarrow x,y \in b_2K$

(g und h sind injektiv bis auf Klebepunkte)

2) $s_Ge \in gB_{1,N},\ e \notin gB_{1,E} \Longrightarrow s_Ge \in gb_1K_N$

$s_He \in hB_{2,N},\ e \notin hB_{2,E} \Longrightarrow s_He \in hb_2K_N$

(Durch die Anwendung von p wird keine Kante von G quell- oder ziellos)

Entsprechende Bedingungen gelten für die Zielfunktionen von G und H.

Im folgenden Satz wird gezeigt, daß die Menge der Σ-Graphen bzgl. Ableitungen über Σ-Produktionen abgeschlossen ist, vorausgesetzt, daß die Wurzel jeder angewendeten Produktion p sich mit keinem anderen Knoten des Ansatzes von p überschneidet.

13 Satz

Sei G Σ-Graph (vgl. I.5), p Σ-Produktion, $r \in R_\Sigma$ (vgl. Prop.6) und

$G \xRightarrow{rp} H$ (vgl. Prop.5) eine direkte Ableitung mit

$$gb_1x_p \neq gb_1x \qquad \text{für alle } x \in K-\{x_p\} . \qquad (1)$$

Dann ist auch H ein Σ-Graph.

Wir zeigen zunächst einige Aussagen, die im Beweis des Satzes benötigt werden.

14 Lemma

Aus (1) folgt $e \in c_1D$ für alle $e \in G_E$ mit $s_Ge \in gb_1(K-\{x_p\})$.

Beweis:

Sei $e \notin c_1D$. Da (g,c_1) Pushoutergänzung von (b_1,d) ist (vgl. Fig.(1) in Def.3), existiert dann $e' \in B_{1,E}-b_1K$ mit $ge' = e$. Daraus folgt nach 9.4 c $s_1e' = b_1x_p$ und, weil g homomorph ist, $gb_1x_p \in gb_1(K-\{x_p\})$. Das widerspricht jedoch (1).

15 Lemma

Aus (1) folgt $e \in c_2D$ für alle $e \in H_E$ mit $s_He \in hb_2(K-\{x_p\})$.

Beweis:

Sei $e \notin c_2D$. Analog zum Beweis von Lemma 14 gibt es dann $e' \in B_{2,E}-b_2K$ mit $he' = e$. Bei $s_2e' = b_2x_p$ erhalten wir $hb_2x_p \in hb_2(K-\{x_p\})$, weil h homomorph ist. Da sich die Injektivität von b_2 (9.2) auf c_2 überträgt und die Diagramme in Def.3 kommutativ sind, folgt daraus $gb_1x_p \in gb_1(K-\{x_p\})$, also ein Widerspruch zu (1). Ist $s_2e' \neq b_2x_p$, dann gilt $s_2e' \notin b_2K$ nach 9.4 d. Klebebedingung 12.1 impliziert $hs_2e' \notin hb_2K$. Dem widerspricht jedoch $hs_2e' = s_He \in hb_2(K-\{x_p\})$.

16 Lemma

$e \in H_E,\ s_H e = hb_2 x_p \Longrightarrow e \in hB_2$.

Beweis:

Sei $e \notin hB_2$. Da (h,c_2) Pushoutergänzung von (b_2,d) ist, existiert dann $e' \in D$ mit $c_2 e' = e$ (2)

und $e' \notin dK$. (3)

(2) impliziert nach Voraussetzung $c_2 s_D e' = c_2 dx_p$, wegen der Injektivität von c_2 $s_D e' = dx_p$, und wir erhalten

$$s_G c_1 e' = gb_1 x_p \ . \tag{4}$$

Aus (3) folgt $c_1 e' \notin c_1 dK = gb_1 K$ nach Prop.10, also

$$c_1 e' \notin gB_1 \tag{5}$$

wegen $c_1 D \cap g(B_1 - b_1 K) = \emptyset$.

Andererseits gilt nach 9.3 und 9.4 a $\text{outdegree}(b_1 x_p) = \text{Rang}(m_1 b_1 x_p) = \text{Rang}(m_G g b_1 x_p) = \text{outdegree}(gb_1 x_p)$, also

$$|s_1^{-1} b_1 x_p| = |s_G^{-1} gb_1 x_p| \ , \tag{6}$$

und $ge = ge'$ mit $s_1 e = b_1 x_p = s_1 e'$ impliziert $m_1 e = m_G ge = m_G ge' = m_1 e'$, also $e = e'$, weil B_1 Σ-Graph ist. Wegen (6) existiert demnach für alle $e \in G_E$ mit $s_G e = gb_1 x_p$ ein $e' \in B_{1,E}$ mit $s_1 e' = b_1 x_p$ und $ge' = e$. Das widerspricht jedoch (4) und (5).

17 Lemma

Aus (1) folgt $s_2 e = b_2 x_p$ für alle $e \in B_{2,E}$ mit $hs_2 e = hb_2 x_p$.

Beweis:

Sei $s_2 e \neq b_2 x_p$. Dann ist wegen der Klebebedingung 12.1 $s_2 e \in b_2(K - \{x_p\})$, also nach Voraussetzung $hb_2 x_p \in hb_2(K - \{x_p\})$. Daraus folgt $gb_1 x_p \in gb_1(K - \{x_p\})$ (vgl. Beweis von Lemma 15), was (1) widerspricht.

Beweis von Satz 13 :

Sei $x \in H_N$. Wir zeigen zunächst $m_H x \in \Sigma \cup C_{var}$ (I.5.1). Für $x \in hB_2$ gilt $m_H x \in rm_2B_2$. Daraus folgt wegen $m_2B_2 \subset \Sigma \cup C_{var}$ und $r \in R_\Sigma$ die Behauptung. $x \in c_2(D\text{-}dK)$ impliziert $m_H x \in m_{D,2}(D\text{-}dK) = m_{D,1}(D\text{-}dK) \subset m_G c_1 D \subset \Sigma \cup C_{var}$ nach Def. der direkten Ableitung und weil G Σ-Graph ist.

Zum Nachweis von I.5.2/3 für x zeigen wir, daß ein Σ-Graph $G_x = (G_{x,N}, G_{x,E}, s_x, t_x, m_x)$ und ein Knoten $y \in G_{x,N}$ mit

$$m_H x = m_x y \,, \tag{7}$$

$$|s_H^{-1} x| = |s_x^{-1} y| \tag{8}$$

und

$$m_H s_H^{-1} x = m_x s_x^{-1} y \tag{9}$$

existieren. Aus (8) folgt outdegree(x) = outdegree(y), und wir erhalten wegen (7) und (9) :

a) $m_H x \in \Sigma \implies$ outdegree$(x) = $ Rang$(m_H x)$,
$m_H s_H^{-1} x = \{1, \ldots, \text{Rang}(m_H x)\}$

b) $mx \in C_{var} \implies$ outdegree$(x) = 0$

1) Sei $x' \in K - \{x_p\}$, $x = hb_2 x'$, $G_x = G$ und $y = gb_1 x'$.

Zu zeigen ist

$$m_H hb_2 x' = m_G gb_1 x' \,, \tag{10}$$

$$|s_H^{-1} hb_2 x'| = |s_G^{-1} gb_1 x'| \tag{11}$$

und

$$m_H s_H^{-1} hb_2 x' = m_G s_G^{-1} gb_1 x' \,. \tag{12}$$

(10) folgt aus 9.4 b. Nach Lemma 14 bzw. 15 gilt $s_G^{-1} gb_1 x' \subset c_1 D$ und $s_H^{-1} hb_2 x' \subset c_2 D$. Da c_1 und c_2 injektiv sind, ist $s_G c_1 e' = gb_1 x'$ genau dann, wenn $s_H c_2 e' = hb_2 x'$ gilt. Damit folgt (11). Aus Prop.11 erhalten wir $m_G c_1 e' = m_H c_2 e'$ für alle $e' \in D_E$, also gilt auch (12).

2) Sei $x = hb_2x_p$, $G_x = B_2$ und $y = b_2x_p$.

Zu zeigen ist $$m_Hhb_2x_p = m_2b_2x_p \;, \tag{13}$$

$$|s_H^{-1}hb_2x_p| = |s_2^{-1}b_2x_p| \tag{14}$$

und $$m_Hs_H^{-1}hb_2x_p = m_2s_2^{-1}b_2x_p \;. \tag{15}$$

(13) folgt aus 9.4 a und $r \in R_{\Sigma}$. Nach Lemma 16 gilt $s_H^{-1}hb_2x_p \subset hB_2$, und nach Lemma 17 ist $s_Hhe = hs_2e = hb_2x_p$ genau dann, wenn $s_2e = b_2x_p$ gilt. Da B_2 Σ-Graph ist, und wegen $m_2e = m_Hhe$ für alle $e \in B_{2,E}$ ist h injektiv auf $s_2^{-1}b_2x_p$. Damit folgt (14) und (15).

3) Sei $x' \in B_2-b_2K$, $x = hx'$, $G_x = B_2$ und $y = x'$.

Zu zeigen ist $$m_Hhx' = m_2x' \;, \tag{16}$$

$$|s_H^{-1}hx'| = |s_2^{-1}x'| \tag{17}$$

und $$m_Hs_H^{-1}hx' = m_2s_2^{-1}x' \;. \tag{18}$$

Da p proper ist, gilt $m_2x' \in C_{fix}$, und es folgt (16). Klebebedingung 12.2 impliziert $s_H^{-1}hx' \subset hB_2$, während aus 12.1 folgt, daß $s_Hhe = hs_2e = hx'$ genau dann gilt, wenn $s_2e = x'$ ist. Da B_2 Σ-Graph ist, und wegen $m_2e = m_Hhe$ für alle $e \in B_{2,E}$ ist h injektiv auf $s_2^{-1}x'$. Damit folgt (17) und (18).

4) Sei $x' \in D-dK$, $x = c_2x'$, $G_x = G$ und $y = c_1x'$.

Zu zeigen ist $$m_Hc_2x' = m_Gc_1x' \;, \tag{19}$$

$$|s_H^{-1}c_2x'| = |s_G^{-1}c_1x'| \tag{20}$$

und $$m_Hs_H^{-1}c_2x' = m_Gs_G^{-1}c_1x' \;. \tag{21}$$

(19) folgt aus $m_{D,1}x' = m_{D,2}x'$ (vgl. Def.3). Wegen $c_1x' \notin gB_1$ gilt $s_G^{-1}c_1x' \cap gB_1 = \emptyset$, also $s_G^{-1}c_1x' \subset c_1D$. Analog folgt $s_H^{-1}c_2x' \subset c_2D$. Da c_1 und c_2 injektiv sind, ist $s_Gc_1e = c_1x'$ genau dann, wenn $s_Hc_2e = c_2x'$

gilt. Damit folgt (20). Aus Prop.11 erhalten wir

$m_G c_1 e = m_H c_2 e$ für alle $e \in D_E$, also gilt auch (21).

Transformationsregeln im Sinne von Def.1 sind als Σ-Produktionen mit Inklusionen b_1 und b_2 darstellbar, und umgekehrt. Diese Beziehung entspricht der Bijektion zwischen Speicherzuständen und Σ-Graphen (vgl. Prop.I.6).

18 Proposition

Sei $t = ((D,P,\mu),(D',P',\eta))$ eine Transformationsregel, $B_1 = Gr(\mu)$, $B_2 = Gr(\eta)$, $K_N = B_{1,N}$, $K_E = B_{1,E} - s_1^{-1} x_t$, s_K und t_K Einschränkungen von s_1 bzw. t_1 auf K_E. Dann ist

$p(t) = (B_1 \xleftarrow{inc} K \xrightarrow{inc} B_2)$ eine Σ-Produktion. p(t) heißt GRAPHREPRÄSENTATION von t. (Bzgl. der Bezeichnungen vgl. Def.2)

Beweis:

Nach Definition ist K Untergraph von B_1. Dasgleiche gilt auch für B_2 : Wegen $D \subset D'$ und $P = P'$ ist $K_N = B_{1,N} \subset B_{2,N}$. Für Kanten $(x,i,y) \in K_E$ gilt $x \neq x_t$ und $(\mu x)_{i+1} = y$. Daraus folgt nach 1.3 $(\eta x)_{i+1} = y$, also $(x,i,y) \in B_{2,E}$. $K_E \subset B_{2,E}$ impliziert $s_K(x,i,y) = s_1(x,i,y) = x = s_2(x,i,y)$ für alle $(x,i,y) \in K_E$.

p(t) ist proper bzgl. R_Σ:

Da C_{var} keine Kantenmarkierungen enthält, ist 7.1-4 nur für Knoten zu zeigen (vgl. Prop.6).

7.1 : Seien $x,y \in B_{1,N}$, $m_1 x = m_1 y \in C_{var} = S \times \omega$. I.6 impliziert $x,y \in P = \{x_1,\dots,x_n\}$. Dann gibt es $1 \leq i,j \leq n$ mit $x_i = x$ und $x_j = y$. Daraus folgt nach I.6.3 $(Typ(x_i),i) = m_1 x_i = m_1 x_j = (Typ(x_j),j)$, also $i = j$

und damit $x = y$.

7.2 : Sei $x \in B_{i,N}$, $m_i x \in C_{var}$. Nach I.6 gilt $x \in P \subset B_{1,N} = K_N$.

7.3 : Sei $x \in K_N$, $m_1 x \in C_{var}$ oder $m_2 x \in C_{var}$. Wegen $x \in P = P' = \{x_1,\dots,x_n\}$ existiert $1 \leq i \leq n$ mit $x_i = x$. Daraus folgt nach I.6.3 $m_1 x_i = (\mathrm{Typ}(x_i),i) = m_2 x_i$, also $m_1 x = m_2 x$.

7.4 : Sei $x \in K_N$, $m_1 x \in C_{fix}$, $(s,i) \in C_{var}$, $r \in R_\Sigma$ mit $r(s,i) = m_1 x$.

a) $x \neq x_t \Longrightarrow m_1 x = (\mu x)_1 = (\eta x)_1 = m_2 x \Longrightarrow$
$m_2 x = r(s,i) \in R_\Sigma(s,i)$.

b) $x = x_t \Longrightarrow s = \mathrm{Sorte}(m_1 x) = \mathrm{Sorte}((\mu x)_1) =$
$\mathrm{Typ}(x) = \mathrm{Sorte}((\eta x)_1) = \mathrm{Sorte}(m_2 x)$ (vgl. Prop.6 und Def.I.2) $\Longrightarrow m_2 x \in \mathrm{Sorte}^{-1}(s) \Longrightarrow$
$r' : C \longrightarrow C$ mit
$$r'c = \begin{cases} m_2 x & \text{falls } c = (s,i) \\ rc & \text{sonst} \end{cases}$$
gehört zu R_Σ. $\Longrightarrow m_2 x = r'(s,i) \in R_\Sigma(s,i)$.

Aus der Definition folgt direkt, daß $p(t)$ fast ist. Nach Prop.I.6 sind B_1 und B_2 Σ-Graphen. Zu zeigen bleibt 9.4. Wähle x_t als Wurzel von $p(t)$.

9.4 a : Wegen $x_t \in D$ gilt $m_1 x_t = (\mu x_t)_1 \in \Sigma$.

9.4 b : Sei $x \in K - \{x_t\}$. Ist $m_1 x \in C_{var}$, dann gilt $m_1 x = m_2 x$, weil $p(t)$ proper ist. $m_1 x \in \Sigma$ impliziert $x \in D$, also $m_1 x = (\mu x)_1 = (\eta x)_1 = m_2 x$ nach 1.3 und I.6.2. $m_1 x \in \omega$ bedeutet, daß x Kante ist. Demnach gibt es $y,z \in B_{1,N}$ mit $x = (y, m_1 x, z)$. Daraus folgt $m_2 x = m_1 x$.

9.4 c : $(x,i,y) \in B_{1,E} - K$ gilt genau dann, wenn $x_t = s_1(x,i,y) = x$ ist.

9.4 d : Sei $(x,i,y) \in B_{2,E}-K$, $x \neq x_t$. Zunächst gilt $(\eta x)_{i+1} = y$, also $x \in D'$. Wäre $x \in K_N = B_{1,N}$, dann würde daraus $x \in D$ folgen. Wegen $x \neq x_t$ wäre $(\mu x)_{i+1} = y$, also $(x,i,y) \in B_{1,E}$. Nach Def. von K_E wäre $(x,i,y) \in K_E$, was der Voraussetzung jedoch widerspricht.

19 Proposition

Sei $p = (B_1 \xleftarrow{\text{inc}} K \xrightarrow{\text{inc}} B_2)$ eine Σ-Produktion. Dann ist $t(p) = (Sz(B_1), Sz(B_2))$ eine Transformationsregel. $t(p)$ heißt ZUSTANDSREPRÄSENTATION von p.

Beweis:

Sei $Sz(B_1) = (D,P,\mu)$ und $Sz(B_2) = (D',P',\eta)$. Da p proper ist, gilt $\{ x \in B_{1,N} = K_N \mid m_1 x \in \Sigma \} \subset \{ x \in B_{2,N} \mid m_2 x \in \Sigma \}$ wegen 7.2/3 ($B_{1,N} = K_N$, weil p fast ist). Daraus folgt $D \subset D'$ nach Prop.I.6. Analog erhält man $P \subset P'$. Wegen 7.2 sind variabel markierte Knoten von $B_{2,N}$ Klebepunkte. Nach 7.3 sind sie deshalb auch in B_1 variabel markiert. Damit gilt $P' \subset P$ nach Prop.I.6.

Sei $x_{t(p)} = x_p$, $x \in D$ mit $x \neq x_p$ und $\mu x = (\sigma, x_1, \ldots, x_n)$. Aus I.6.8 folgt $m_1 x = \sigma$, und I.6.9 impliziert, daß für alle $1 \leq i \leq n$ eine Kante $e_i \in B_{1,E}$ mit $s_1 e_i = x$, $t_1 e_i = x_i$ und $m_1 e_i = i$ existiert. Wegen $x \neq x_p$ ist $m_2 x = \sigma$ nach 9.4b und $e_i \in K$ nach 9.4c. Da die Inklusion $K \subset B_2$ quell- und zielverträglich und nach 9.4b auch markierungsverträglich ist, folgt $s_2 e_i = x$, $t_2 e_i = x_i$ und $m_2 e_i = i$ für alle $1 \leq i \leq n$. Nach I.6.8/9 gilt dann $\eta x = (\sigma, x_1, \ldots, x_n)$.

Kapitel III

DIE KORREKTHEIT VON TRANSFORMATIONSREGELN

In Kapitel I haben wir die funktionale Semantik von Speicherzuständen definiert (I.13) und im vorangegangenen Kapitel Zustandstransformationen als Ableitungen über einer Graph-Grammatik beschrieben. Nun liegt es nahe, nach der Korrektheit solcher Ableitungen zu fragen, d.h. ob sie die Semantik der Speicherzustände bewahren.

Wir bilden für jedes Paar $t = (\mu, \eta)$ eines Systems T von Transformationsregeln die Baumrepräsentationen

$$L[t] = Br(\mu)(x_t) \text{ und } R[t] = Br(\eta)(x_t)$$

von μ und η an der Stelle x_t (vgl. I.19). Ist A eine ω-stetige Σ-Algebra derart, daß für alle $t \in T$ die Interpretationen L[t,A] und R[t,A] von L[t] bzw. R[t] in A gleich sind, dann bleibt die funktionale Semantik von Speicherzuständen bzgl. A bei Ableitungen über T erhalten.

Dazu wird zunächst gezeigt, daß L[t,A] und R[t,A] den kleinsten Lösungen von $\tilde{\mu}$ bzw. $\tilde{\eta}$ in A an der Stelle x_t entsprechen. Im zweiten Schritt wird die Korrektheit von T bzgl. A mit Hilfe der in Kapitel I gezeigten Semantikverträglichkeit von Σ-Graphmorphismen (I.24) auf die Korrektheit von Ableitungen über T fortgesetzt.

Als Vorbereitung auf Kapitel IV wiederholen wir die Begriffe Church-Rosser-System und Normalform, aus denen sich für Regelsysteme zur Auswertung von Ausdrücken der operationelle Semantikbegriff herleitet. Daraus ergeben sich einige Überlegungen zur Vollständigkeit solcher Regelsysteme.

1 <u>Definition</u> (vgl. I.8-15)

Sei $t \in CT_{\Sigma(Y)}$ und $t^{(n)}$ die Einschränkung von t auf $\bigcup_{i<n} \omega^i$. $t^{(n)}$ läßt sich als Term in $T_{\Sigma \cup \{\perp\}}(Y)$ auffassen, bei dem die "abgeschnittenen Unterbäume" von t durch das Symbol $\perp$ ersetzt sind. Sei A eine ω-stetige Σ-Algebra, $b \in A^Y$ und b^* die eindeutige Σ-homomorphe Fortsetzung von b auf $T_{\Sigma \cup \{\perp\}}(Y)$ mit $b^* \perp = \perp_A$. Dann ist

$$b^+ : CT_{\Sigma(Y)} \longrightarrow A$$

definiert durch $\quad b^+ t = \bigsqcup_{n \in \omega} b^* t^{(n)}$.

2 <u>Proposition</u> (/GTWW 77/, Th.4.8)

b^+ ist die eindeutige ω-stetige und Σ-homomorphe Fortsetzung von b auf $CT_{\Sigma(Y)}$.

3 <u>Proposition</u>

Begreift man A als $\Sigma(Y)$-Algebra A(b) (vgl. I.12), dann ist b^+ $\Sigma(Y)$-homomorph.

Beweis:

Nach Prop.2 bleibt nur die Y-Verträglichkeit von b^+ zu zeigen. Für alle $y \in Y$ gilt $b^+ y_{CT} = b^+ y = b^* y = by = y_{A(b)}$. Daraus folgt die Behauptung.

Das folgende Lemma erweitert /GTWW 77/,Prop.5.1 auf parametrisierte Gleichungssysteme und besagt, bezogen auf Gleichungen im Sinne von Prozedurdefinitionen, daß der kleinste Fixpunkt eines rekursiven Programms P der Interpretation der Expansion von P entspricht. Dieses Ergebnis wurde auf unterschiedlichen Wegen auch in den Arbeiten von Nivat (/Niv 74/) und Arbib (/Arb 77/) erzielt und dort als Äquivalenz zwischen funktionaler und operationeller Semantik interpretiert.

4 <u>Lemma</u>

Sei $D \subset X$, $E: D \longrightarrow T_{\Sigma(Y)}(D)$ ein parametrisiertes Gleichungssystem und $b \in A^Y$. Dann gilt

$$b^+ \circ \left| E_{CT_{\Sigma(Y)}} \right| = \left| E_{A(b)} \right| \quad .$$

Beweis:

Sei $\bar{E} = E_{CT_{\Sigma(Y)}}$. Wir zeigen zunächst für alle $f \in CT^D_{\Sigma(Y)}$

$$b^+ \circ (\bar{E} f) = E_{A(b)}(b^+ \circ f) \quad . \tag{1}$$

Sei $f \in CT^D_{\Sigma(Y)}$ und f^* die eindeutige $\Sigma(Y)$-homomorphe Fortsetzung von f auf $T_{\Sigma(Y)}(D)$. Dann ist das Diagramm (2) kommutativ.

$$\begin{tikzcd}
D \arrow[rr, "inc"] \arrow[dr, "f"'] & {} \arrow[d, phantom, "(2)"] & T_{\Sigma(Y)}(D) \arrow[dl, "f^*"] \\
& CT_{\Sigma(Y)} \arrow[d, "b^+"] & \\
& A(b) &
\end{tikzcd}$$

Da auch für alle $g \in A(b)^D$ eine eindeutige $\Sigma(Y)$-homomorphe Fortsetzung g^* von g auf $T_{\Sigma(Y)}(D)$ existiert, folgt aus Prop.2

$$(b^+ \circ f)^* = b^+ \circ f^* \quad .$$

Damit gilt nach Definition von $\bar{E}$ und $E_{A(b)}$ (vgl. I.11) :

$b^+ \circ (\bar{E} f) = b^+ \circ (f^* \circ E) = (b^+ \circ f^*) \circ E = (b^+ \circ f)^* \circ E = E_{A(b)}(b^+ \circ f)$,

also (1). Daraus erhält man durch Induktion über n :

$$b^+ \circ (\bar{E}^n(\perp)) = E^n_{A(b)}(\perp) \quad . \tag{3}$$

Für $n = 0$ folgt (3) aus $b^+ \perp = \perp_A$ (vgl. Def.1).

Sei (3) für n erfüllt.

Dann gilt wegen (1) $b^{+}\circ(\overline{E}^{n+1}(\perp)) = b^{+}\circ(\overline{E}(\overline{E}^{n}(\perp))) =$
$E_{A(b)}(b^{+}\circ(\overline{E}^{n}(\perp))) = E_{A(b)}(E^{n}_{A(b)}(\perp)) = E^{n+1}_{A(b)}(\perp)$.
Aus (3) folgt schließlich $b^{+}\circ|\overline{E}| = b^{+}\circ\bigsqcup_{n\in\omega}\overline{E}^{n}(\perp) =$
$\bigsqcup_{n\in\omega} b^{+}\circ(\overline{E}^{n}(\perp)) = \bigsqcup_{n\in\omega} E^{n}_{A(b)}(\perp) = |E_{A(b)}|$, da b^{+} nach Prop.2 ω-stetig ist.

5 <u>Definition</u>

Sei $\mathcal{E} \subset CT^{2}_{\Sigma}(Y)$. Eine ω-stetige Σ-Algebra A heißt Σ-$\mathcal{E}$-ALGEBRA, wenn für alle $(L,R) \in \mathcal{E}$ und $b \in A^{Y}$

$$b^{+}L = b^{+}R$$

gilt.

6 <u>Definition</u>

Sei T eine Menge von Transformationsregeln.
$\mathcal{E}_{T} = \{\ (Br(\mu)(x_t), Br(\eta)(x_t))\ |\ t = (\mu,\eta) \in T\ \}$
ist die Menge der TRANSFORMATIONSGLEICHUNGEN von T.

7 <u>Voraussetzung</u> (für 8-11)

Sei T eine Menge von Transformationsregeln, A eine Σ-$\mathcal{E}_{T}$-Algebra, $t = ((D,P,\mu),(D',P',\eta)) \in T$, G und H Σ-Graphen, $Sz(G) = (D_1,P_1,\alpha)$, $Sz(H) = (D_2,P_2,\beta)$, $r \in R_{\Sigma}$ und $G \xRightarrow{rp(t)} H$.

Da p(t) fast ist, ist c_1 bijektiv auf Knoten und c_2 injektiv. Folglich ist die "Restabbildung"

$$f = c_{2,N}c^{-1}_{1,N} : G_N \longrightarrow H_N$$

wohldefiniert und injektiv. Bzgl. der Bezeichnungen von Mengen und Abbildungen einer direkten Ableitung vgl. Def. II.3.

8 <u>Lemma</u>

Die Einbettung $g:B_1 \longrightarrow G$ ist ein Σ-Graphmorphismus (vgl. I.5). (Dasgleiche gilt für $h:B_2 \longrightarrow H$.)

Beweis:

1) $m_1x \in C_{fix} \Longrightarrow m_1x = rm_1x = m_Ggx$.

2) $m_1x \in C_{var} \Longrightarrow m_Ggx = rm_1x \in \mathrm{Sorte}^{-1}((m_1x)_1) \cup \{m_1x\}$ wegen $r \in R_\Sigma$ (vgl. II.6) $\Longrightarrow m_Ggx = m_1x$ oder $m_Ggx \in \Sigma$ und $\mathrm{Sorte}(m_Ggx) = (m_1x)_1$.

9 <u>Lemma</u>

$$x \in P_1 \Longleftrightarrow fx \in P_2 \quad .$$

Beweis:

1) Sei $x \in gB_{1,N}$. Dann existiert $x' \in B_{1,N} = K_N$ mit $gx' = x$. Wegen $gx' = gb_1x' = c_1dx'$ gilt $fx = c_2c_{1,N}^{-1}gx' = c_2dx' = hb_2x' = hx'$. Demnach bleibt zu zeigen

$$gx' \in P_1 \Longleftrightarrow hx' \in P_2 \quad . \tag{1}$$

Sei $gx' \in P_1$. Dann gilt $x' \in P$ nach Prop.I.7, also $m_2x' = m_1x' \in C_{var}$, da p(t) proper ist. Daraus folgt $m_Hhx' = rm_2x' = rm_1x' = m_Ggx' \in C_{var}$ wegen $gx' \in P_1$, so daß $hx' \in P_2$ ist. Analog zeigt man die andere Richtung von (1).

2) Sei $x \in c_1(D_N - dK)$. Dann existiert $x' \in D_N - dK$ mit $c_1x' = x$. Wegen $fx = c_2c_{1,N}^{-1}c_1x' = c_2x'$ bleibt zu zeigen

$$c_1x' \in P_1 \Longleftrightarrow c_2x' \in P_2 \quad .$$

Sei $c_1x' \in P_1$. Daraus folgt $m_Hc_2x' = m_{D,2}x' = m_{D,1}x' = m_Gc_1x' \in C_{var}$, also $c_2x' \in P_2$. Ebenso gilt die Umkehrung.

10 Lemma

Sei $c \in A^Y$ mit $cfy = cy$ für alle $y \in P_1$. (1)

Dann gilt für alle $x \in D_1$

$$|\tilde{\alpha}_{A(c)}|x = |\tilde{\beta}_{A(c)}|fx \quad .$$

Beweis:

Aus Lemma 4 folgt für alle $b \in A^Y$

$$|\tilde{\mu}_{A(b)}|(x_t) = b^+ \circ |\tilde{\mu}_{CT_{\Sigma(Y)}}|(x_t) = b^+ \circ Br(\mu)(x_t) =$$

$$b^+ \circ Br(\eta)(x_t) = b^+ \circ |\tilde{\eta}_{CT_{\Sigma(Y)}}|(x_t) = |\tilde{\eta}_{A(b)}|(x_t) \quad . \qquad (2)$$

Wir zeigen zunächst für alle $x \in D_1$

$$\tilde{\alpha}^n_{A(c)}(\perp)x \leqslant |\tilde{\beta}_{A(c)}|fx \qquad (3)$$

durch Induktion über n.

$n = 0$ ist klar. Sei (3) für n erfüllt.

1) Sei $gx_t \neq x \in D_1$ und $\alpha x = (\sigma, x_1, \ldots, x_n)$.

Nach II.9.4 c gilt $s_G^{-1}x \subset c_1 D_E$. Folglich ist

$f' : \{x\} \cup s_G^{-1}x \cup t_G s_G^{-1}x \longrightarrow H$ mit $f'y = c_2 c_1^{-1} y$

für alle $y \in Def(f')$ wohldefiniert, so daß man wie

im Beweis von Prop.I.21 wegen II.9.4 b

$\beta fx = (\sigma, fx_1, \ldots, fx_n)$ folgern kann. Da $|\tilde{\beta}_{A(c)}|$

Fixpunkt von $\tilde{\beta}_{A(c)}$ ist (vgl. Prop.I.14), erhalten

wir nach Induktionsvoraussetzung

$$\tilde{\alpha}^{n+1}_{A(c)}(\perp)x = \tilde{\alpha}_{A(c)}(\tilde{\alpha}^n_{A(c)}(\perp))x = \tilde{\alpha}^n_{A(c)}(\perp)^* \tilde{\alpha} x$$

$$= \sigma_A(\tilde{\alpha}^n_{A(c)}(\perp)x_1, \ldots, \tilde{\alpha}^n_{A(c)}(\perp)x_n)$$

$$\leqslant \sigma_A(|\tilde{\beta}_{A(c)}|fx_1, \ldots, |\tilde{\beta}_{A(c)}|fx_n) = |\tilde{\beta}_{A(c)}|fx \quad .$$

2) Sei $\mu x_t = (\sigma, x_1, \ldots, x_n)$. (4)

Dann ist nach Lemma 8 und Prop. I.21

$$\alpha g x_t = (\sigma, gx_1, \ldots, gx_n) \quad . \qquad (5)$$

Ferner seien $u, v, w \in A^Y$ definiert durch

$$uy = \begin{cases} \tilde{\alpha}^n_{A(c)}(\perp)gy & \text{falls } y \in P \text{ und } gy \in D_1 \\ cgy & \text{falls } y \in P \text{ und } gy \in P_1 \\ cy & \text{sonst} \end{cases}$$

$$vy = \begin{cases} |\tilde{\beta}_{A(c)}|^{fgy} & \text{falls } y \in P \text{ und } fgy \in D_2 \\ cfgy & \text{falls } y \in P \text{ und } fgy \in P_2 \\ cy & \text{sonst} \end{cases}$$

$$wy = \begin{cases} |\tilde{\beta}_{A(c)}|^{hy} & \text{falls } y \in P' \text{ und } hy \in D_2 \\ chy & \text{falls } y \in P' \text{ und } hy \in P_2 \\ cy & \text{sonst} \end{cases}$$

Nach Induktionsvoraussetzung, (1) und Lemma 9 ist

$$u \leqslant v \quad . \tag{6}$$

Wegen $P = P'$ und $fgx = hx$ für alle $x \in B_{1,N}$ gilt

$$v = w \quad . \tag{7}$$

Damit erhält man

$$\begin{aligned}
&\tilde{\alpha}^{n+1}_{A(c)}(\perp)gx_t \\
&= \sigma_A(\tilde{\alpha}^n_{A(c)}(\perp)gx_1, \ldots, \tilde{\alpha}^n_{A(c)}(\perp)gx_n) && \text{(nach (5))} \\
&\leqslant \sigma_A(\tilde{\mu}^n_{A(u)}(\perp)x_1, \ldots, \tilde{\mu}^n_{A(u)}(\perp)x_n) && \text{(nach Lemma I.23)} \\
&\leqslant \sigma_A(\tilde{\mu}^n_{A(v)}(\perp)x_1, \ldots, \tilde{\mu}^n_{A(v)}(\perp)x_n) && \text{(nach (6))} \\
&= \tilde{\mu}^{n+1}_{A(v)}(\perp)x_t && \text{(nach (4))} \\
&\leqslant |\tilde{\mu}_{A(v)}|^{x_t} \\
&= |\tilde{\eta}_{A(v)}|^{x_t} && \text{(nach (2))} \\
&= |\tilde{\eta}_{A(w)}|^{x_t} && \text{(nach (7))} \\
&= |\tilde{\beta}_{A(c)}|^{hx_t} && \text{(nach Lemma 8 und Satz I.24)} \\
&= |\tilde{\beta}_{A(c)}|^{fgx_t} \quad .
\end{aligned}$$

Aus (3) folgt für alle $x \in D_1$

$$|\tilde{\alpha}_{A(c)}|x = \bigsqcup_{n\in\omega} \tilde{\alpha}^n_{A(c)}(\perp)x \leqslant |\tilde{\beta}_{A(c)}|fx \ . \tag{8}$$

Analog zeigt man $|\tilde{\beta}_{A(c)}|fx \leqslant |\tilde{\alpha}_{A(c)}|x$. (9)

(8) und (9) ergeben die Behauptung.

11 Definition

$c \in A^Y$ heißt VERTRÄGLICH mit der Restabbildung f, wenn für alle $y \in P_1$ $cfy = cy$ gilt.

12 Satz

Sei T eine Menge von Transformationsregeln, A eine Σ-$\mathcal{E}_T$-Algebra, $\{t_i\}_{1\leqslant i\leqslant n} \subset T$, $\{G_i\}_{0\leqslant i\leqslant n}$ eine Menge von Σ-Graphen, $Sz(G_0) = (D_1, P_1, \alpha)$, $Sz(G_n) = (D_2, P_2, \beta)$, $\{r_i\}_{1\leqslant i\leqslant n} \subset R_\Sigma$,

$$a = \{ G_{i-1} \xRightarrow{r_i p(t_i)} G_i \}_{1\leqslant i\leqslant n}$$

eine Ableitung von G_0 nach G_n und $\{f_i\}_{1\leqslant i\leqslant n}$ die Menge der zu a gehörigen Restabbildungen (vgl. 7). Sei $c \in A^Y$ mit allen f_i verträglich. Dann gilt für alle $x \in D_1$

$$|\tilde{\alpha}_{A(c)}|x = |\tilde{\beta}_{A(c)}|f_n \ldots f_1 x \ .$$

Beweis:

Durch Induktion über n erhält man die Behauptung aus Lemma 10.

13 Notation

Seien G,H Graphen und P eine Graph-Grammatik. Gibt es eine Ableitung von G nach H über P, dann schreiben wir

$$G \underset{P}{\overset{*}{\Longrightarrow}} H \ .$$

14 Definition

Sei $\mathcal{G}$ eine Menge von Graphen und P eine Graph-Grammatik. G ε $\mathcal{G}$ ist in NORMALFORM bzgl. $\mathcal{G}$ und P, wenn kein p ε P und kein H ε $\mathcal{G}$ mit

$$G \overset{p}{\Longrightarrow} H$$

existiert. Für alle Graphen G' mit $G' \overset{*}{\underset{P}{\Longrightarrow}} G$ ist G eine Normalform von G'.

P heißt CHURCH-ROSSER-SYSTEM bzgl. $\mathcal{G}$ (P ist CR), wenn es für alle G,H,H^* ε $\mathcal{G}$ mit $G \overset{*}{\underset{P}{\Longrightarrow}} H$ und $G \overset{*}{\underset{P}{\Longrightarrow}} H^*$ ein X ε $\mathcal{G}$ mit $H \overset{*}{\underset{P}{\Longrightarrow}} X$ und $H^* \overset{*}{\underset{P}{\Longrightarrow}} X$ gibt.

15 Proposition

Ist P CR bzgl. $\mathcal{G}$, dann existiert für alle G ε $\mathcal{G}$ höchstens eine Normalform bzgl. $\mathcal{G}$ und P.

Sei $\mathcal{G}$ eine Menge von Σ-Graphen und P eine Σ-Grammatik zur Auswertung der durch die Elemente von $\mathcal{G}$ repräsentierten Ausdrücke. Existiert eine eindeutige Normalform von G ε $\mathcal{G}$, so liegt es nahe, sie als OPERATIONELLE SEMANTIK von G zu bezeichnen.

In diesem Zusammenhang stellt sich die Frage nach der VOLLSTÄNDIGKEIT von P bzgl. einer "semantischen" Algebra A : ob alle bzgl. A semantisch äquivalenten Σ-Graphen eine gemeinsame Normalform haben. Dies ist quasi die Umkehrung der oben behandelten Semantikverträglichkeit von Ableitungen über P. Besteht P aus den Graphrepräsentationen einer Menge T von Transformationsregeln, so müßte P genau dann vollständig sein, wenn A eine "freie Σ-Struktur" ist, die gerade den Transformationsgleichungen $\mathcal{E}_T$ genügt, und P ein Church-Rosser-System ist.

Eine solche Algebra ist der nach dem kongruenten Abschluß $\equiv_T$ von $\mathcal{E}_T$ gebildete Quotient $CT_\Sigma/\equiv_T$ von CT_Σ. Den Paaren von Σ-Bäumen in $\equiv_T$ entsprechen Paare von Σ-Graphen in der symmetrischen Hülle der Ableitungen über P. Ist P CR, dann haben die beiden Graphen eines solchen Paares eine gemeinsame Normalform.

Der Nachweis, daß eine vorgegebene "semantische" Algebra zu $CT_{\Sigma}/\equiv_T$ isomorph ist, kann möglicherweise durch den Umweg über "kanonische Baumalgebren" erbracht werden, ähnlich wie die Korrektheit von Datentypspezifikationen mit Hilfe kanonischer Termalgebren gezeigt wird (vgl. /GTW 76/,Kap.4.1).

Kapitel IV

CHURCH-ROSSER-EIGENSCHAFTEN

Wir stellen einige Sätze über die Vertauschbarkeit von Ableitungssequenzen vor, die sich im wesentlichen auf drei Ergebnisse aus der Theorie der Graph-Grammatiken stützen :

1) Systeme proprer Produktionen, deren linke Seiten sich nur in Klebepunkten überschneiden können, sind Church-Rosser, falls eine Markierungsbedingung erfüllt ist (/ER 76/,Th.5.9).
2) "Lokale" Ableitungssequenzen lassen sich in "globale" einbetten: Gilt $G \overset{*}{\Longrightarrow} H$ und ist $h:G \longrightarrow \overline{G}$ ein Graphmorphismus, dann gilt auch $\overline{G} \overset{*}{\Longrightarrow} \overline{H}$, wobei $\overline{H}$ - grob gesagt - die Verklebung von H mit $\overline{G}$-hG ist (/EK 76/,Th.4).
3) Garbage collection als Beseitigung unreferenzierter Knoten ist mit der Anwendung von fast und TREELIKE Produktionen vertauschbar (/ER 77/,Th.6.4).

Bzgl. Grammatiken zur Auswertung graphrepräsentierter Ausdrücke hat ein Church-Rosser-System den Vorzug, daß das Resultat der Auswertung unabhängig von der verwendeten Strategie ist (vgl. Prop.III.15). In diesem Fall können die Transformationsschritte nebenläufig, d.h. in nicht festgelegter Reihenfolge, bei PARALLELER UNABHÄNGIGKEIT der Produktionen sogar parallel ausgeführt werden (vgl. /EK 75/).

Wir nennen eine Graph-Grammatik P schwach Church-Rosser, wenn es für je zwei direkte Ableitungen der Form

$$G \overset{p}{\Longrightarrow} H \qquad \text{bzw.} \qquad G \overset{\overline{p}}{\Longrightarrow} \overline{H}$$

mit $p,\overline{p} \in P$ zwei in H bzw. $\overline{H}$ beginnende konvergierende Ableitungssequenzen gibt. Eine Σ-Grammatik P wird als schwach unabhängig bezeichnet, wenn für alle $p,\overline{p} \in P$ höchstens eine Verklebung $V(p,\overline{p})$ der linken Seiten von p und $\overline{p}$ in Nichtklebepunkten existiert und die Wurzeln von p und $\overline{p}$ mit keinem anderen Knoten von $V(p,\overline{p})$ identifizierbar sind.

Da Σ-Produktionen proper sind, erhalten wir mit Hilfe von 1) und 2) ein Kriterium für die schwache Church-Rosser-

Eigenschaft einer schwach unabhängigen Σ-Grammatik P: Hinreichend ist, daß für alle $p,\overline{p} \in P$ zwei in H_p bzw. $H_{\overline{p}}$ beginnende konvergierende Ableitungssequenzen existieren, die zwei leicht nachprüfbare Bedingungen erfüllen. H_p und $H_{\overline{p}}$ sind dabei durch

$$V(p,\overline{p}) \xrightarrow{p} H_p \qquad \text{bzw.} \qquad V(p,\overline{p}) \xrightarrow{\overline{p}} H_{\overline{p}}$$

definiert. Mit diesem Kriterium zeigen wir, daß Systeme von ZEIGERUMSETZENDEN, UNZUSAMMENHÄNGENDEN Σ-Produktionen und COLLAPSE-REGELN schwach Church-Rosser sind.

Ferner führen wir als eine hinreichende Bedingung für ihre parallele Unabhängigkeit die BÜSCHELDISJUNKTHEIT zweier Produktionen p und $\overline{p}$ ein, die sich durch lokalen Vergleich der linken Seiten von p und $\overline{p}$ nachprüfen läßt.

Schließlich wird 3) auf Produktionen erweitert, die selbst Knoten beseitigen, also nicht fast sind.

Die beiden letzten Ergebnisse gehen auf eine frühere Arbeit (/Pad 77/) zurück, in der wir die Komponenten eines LISP-Interpreters im Gegensatz zur Version in Kapitel V als büscheldisjunkte Produktionen, die nicht fast sind, dargestellt haben. Die korrekte Auswertung von LISP-Ausdrücken mit Hilfe jener Produktionen macht es jedoch erforderlich, daß die Grammatik um Substitutionsregeln für alle im auszuwertenden Ausdruck vorkommenden rekursiven Definitionen (LABEL-Terme) "dynamisch" erweitert wird.

1 <u>Definition</u>

Sei P eine Graph-Grammatik und R eine Menge von recolorings. $p, \overline{p} \in P$ heißen PARALLEL UNABHÄNGIG, wenn für je zwei natürliche direkte Ableitungen

$$G \xRightarrow{rp} H \quad \text{und} \quad G \xRightarrow{\overline{rp}} \overline{H}$$

mit $r, \overline{r} \in R$

$$gB_1 \cap \overline{gB}_1 \subset gb_1K \cap \overline{gb}_1\overline{K}$$

gilt. (Die Bedeutung der Bezeichnungen für Mengen und Abbildungen einer direkten Ableitung entnehme man II.2/3.)

2 <u>Definition</u>

Seien P und R wie oben. $p, \overline{p} \in P$ heißen MARKIERUNGSKONSISTENT, wenn für je zwei natürliche direkte Ableitungen

$$G \xRightarrow{rp} H \quad \text{und} \quad G \xRightarrow{\overline{rp}} \overline{H}$$

mit $r, \overline{r} \in R$ und für alle $x \in K$, $\overline{x} \in \overline{K}$ mit

$$m_1b_1x,\ \overline{m}_1\overline{b}_1\overline{x} \in C_{fix} \text{ und } gb_1x = \overline{gb}_1\overline{x}$$

$m_1b_1x = m_2b_2x$ und $\overline{m}_1\overline{b}_1\overline{x} = \overline{m}_2\overline{b}_2\overline{x}$ gilt.

3 <u>Definition</u>

Seien P und R wie oben. P heißt STARK CHURCH-ROSSER, falls für alle $p, \overline{p} \in P$ und je zwei natürliche direkte Ableitungen

$$G \xRightarrow{rp} H \quad \text{und} \quad G \xRightarrow{\overline{rp}} \overline{H}$$

mit $r, \overline{r} \in R$ ein Graph X und zwei recolorings $s, \overline{s} \in R$ mit

$$H \xRightarrow{\overline{sp}} X \quad \text{und} \quad \overline{H} \xRightarrow{sp} X$$

existieren.

4 <u>Satz</u> (/ER 76/,Th.5.9)

Sei R eine vollständige Menge von recolorings. Dann ist jede Graph-Grammatik aus parallel unabhängigen, markierungskonsistenten und propren Produktionen stark Church-Rosser.

Sei $\{ G_i = (G_{i,N}, G_{i,E}, s_i, t_i, m_i) \}_{0 \leqslant i \leqslant n}$ eine Menge markierter Graphen und

$$\{ a_i = (G_{i-1} \xRightarrow{p_i} G_i) \}_{1 \leqslant i \leqslant n} \qquad (1)$$

eine Folge natürlicher direkter Ableitungen, wobei a_i durch die beiden Pushouts

$$\begin{array}{ccccccc} B_{1i} & \xleftarrow{b_{1i}} & K_{1i} & \quad & K_{2i} & \xrightarrow{b_{2i}} & B_{2i} \\ {\scriptstyle g_{1i}}\downarrow & & \downarrow{\scriptstyle d_i} & & {\scriptstyle d_i}\downarrow & & \downarrow{\scriptstyle g_{2i}} \\ G_{i-1} & \xleftarrow[c_{1i}]{} & D_{1i} & & D_{2i} & \xrightarrow[c_{2i}]{} & G_i \end{array} \qquad (2)$$

gegeben ist. Sei $h: G_0 \longrightarrow \overline{G}_0$ ein markierungsverträglicher Graphmorphismus. Der von den Mengen

$$\{ x \in G_{0,N} \mid \text{es gibt } e \in \overline{G}_{0,E} - hG_0 \text{ mit } \overline{s}_0 e = hx \text{ oder } \overline{t}_0 e = hx \}$$

und

$$\{ x \in G_0 \mid \text{es gibt } y \in G_0 \text{ mit } y \neq x \text{ und } hx = hy \}$$

erzeugte Untergraph von G_0 heißt RAND (Rd) von G_0 in $\overline{G}_0$. Nach /Ehr 78/,3.5 existiert ein eindeutiges PUSHOUTKOMPLEMENT $Rd \xrightarrow{\alpha} \overline{Rd} \xrightarrow{\overline{f}_0} \overline{G}_0$ von $Rd \xrightarrow{f_0} G_0 \xrightarrow{h} \overline{G}_0$, wobei f_0 die Inklusion von Rd in G_0 ist.

Die KRITISCHEN PUNKTE $Krit_i$ von G_i werden induktiv durch $Krit_n = \emptyset$ und $Krit_{i-1} = c_{1i} c_{2i}^{-1} Krit_i \cup (g_{1i} B_{1i} - g_{1i} b_{1i} K_i)$ definiert (vgl. Fig.~~(1)~~ 2)). Diese Voraussetzungen führen zum

5 Einbettungssatz für natürliche Ableitungen (/EK 76/,Th.4)

Sei $Rd \cap Krit_0 = \emptyset$. Dann ist $f_i = c_{2i} c_{1i}^{-1} f_{i-1}$ für alle $1 \leqslant i \leqslant n$ wohldefiniert. ($f_i x$ heißt RESIDUE von $x \in Rd$ in G_i.) Gilt außerdem $m_i f_i x = m_i f_i y$ für alle $x,y \in Rd$ mit $\alpha x = \alpha y$, dann erhalten wir durch die Pushoutergänzungen $\overline{Rd} \xrightarrow{\overline{f}_i} \overline{G}_i \xleftarrow{h_i} G_i$ von $\overline{Rd} \xleftarrow{\alpha} Rd \xrightarrow{f_i} G_i$ eine Folge

$$\{ \overline{a}_i = (\overline{G}_{i-1} \xRightarrow{p_i} \overline{G}_i) \}_{1 \leq i \leq n}$$

natürlicher direkter Ableitungen.

Satz 5 gilt auch für Graphen G_0 mit variablen Markierungen und für Graphmorphismen $h: G_0 \longrightarrow \overline{G}_0$ mit $rm_0 = \overline{m}_0 h$, wobei r ein beliebiges recoloring ist. Die Markierungsbedingung in Satz 5 ist in diesem Fall durch

$$x,y \in Rd,\ \alpha x = \alpha y \quad \Longrightarrow \quad rm_i f_i x = rm_i f_i y \quad (1 \leq i \leq n) \qquad (3)$$

zu ersetzen. Nach Konstruktion von $\overline{G}_i$ gibt es dann eine eindeutige Markierungsfunktion $\overline{m}_i$ von $\overline{G}_i$ mit $rm_i = \overline{m}_i h_i$ und $\overline{m}_0 \overline{f}_0 x = \overline{m}_i \overline{f}_i x$ für alle $x \in \overline{Rd} - \alpha Rd$.

(vgl. /ED 75/,3.10 : Konstruktion markierter Verklebungen)

6 Proposition

Die Markierungsbedingung (3) zur Anwendung von Satz 5 ist erfüllt, wenn alle direkten Ableitungen a_i (vgl. (1)) die Markierungen durch h identifizierter Randpunkte bewahren, d.h. für alle $x,y \in Rd$ mit $hx = hy$ gilt $m_{i-1} f_{i-1} x = m_i f_i x$ und $m_{i-1} f_{i-1} y = m_i f_i y$ für alle $1 \leq i \leq n$.

Beweis durch Induktion über i :
Seien $x,y \in Rd$ mit $hx = hy$. Daraus folgt $hf_0 x = hf_0 y$, also $rm_0 f_0 x = \overline{m}_0 hf_0 x = \overline{m}_0 hf_0 y = rm_0 f_0 y$.
Sei (3) für $0 \leq i < n$ erfüllt. Dann gilt (3) nach Voraussetzung auch für i+1.

Das folgende Korollar verleiht dem Einbettungssatz entscheidende Bedeutung im Hinblick auf den Nachweis von Church-Rosser-Eigenschaften.

7 Korollar

Seien $\{ G_{i-1} \xRightarrow{p_i} G_i \}_{1 \leq i \leq n}$ und $\{ G'_{i-1} \xRightarrow{p'_i} G'_i \}_{1 \leq i \leq m}$

zwei Folgen natürlicher direkter Ableitungen mit $G_0 = G'_0$, $G_n = G'_m$ und $h:G_0 \longrightarrow \overline{G}_0$ ein (bis auf recoloring) markierungsverträglicher Graphmorphismus. Dann gibt es unter den Vor. von Satz 5 "Restabbildungen" f_i für alle $1 \leqslant i \leqslant n$, f'_i für alle $1 \leqslant i \leqslant m$ sowie zwei Folgen

$$\{ \overline{G}_{i-1} \xRightarrow{p_i} \overline{G}_i \}_{1 \leqslant i \leqslant n} \quad \text{und} \quad \{ \overline{G}'_{i-1} \xRightarrow{p'_i} \overline{G}'_i \}_{1 \leqslant i \leqslant m}$$

natürlicher direkter Ableitungen.

$f_n = f'_m$ impliziert $\overline{G}_n = \overline{G}'_m$. (4)

Beweis:

(4) folgt aus der Konstruktion von $\overline{G}_n$ und $\overline{G}'_m$ (vgl. Satz 5).

8 Definition

Sei P eine Graph-Grammatik und $\mathcal{G}$ eine Menge markierter Graphen. P heißt SCHWACH CHURCH-ROSSER bzgl. $\mathcal{G}$, falls für alle $p,\overline{p} \in P$ und je zwei direkte Ableitungen

$$G \xRightarrow{p} H \quad \text{und} \quad G \xRightarrow{\overline{p}} \overline{H}$$

mit $G,H,\overline{H} \in \mathcal{G}$ ein Graph $X \in \mathcal{G}$ mit

$$H \xRightarrow[P]{*} X \quad \text{und} \quad \overline{H} \xRightarrow[P]{*} X$$

existiert.

9 Definition

Eine Σ-Grammatik P heißt SCHWACH UNABHÄNGIG, wenn für alle Paare $(p,\overline{p}) \in P \times P$, Σ-Graphen G und je zwei verschied. Ableitungen $G \xRightarrow{p} H$ und $G \xRightarrow{\overline{p}} \overline{H}$

die folgenden Bedingungen erfüllt sind (vgl. II.9) :

1) $gb_1 x_p \neq gb_1 x$ für alle $x \in K - \{x_p\}$.
 (vgl. II.13(1))

2) Es existiert ein SCHNITTPUNKT $\ddot{u}[p,\overline{p}] \in B_{1,N}$ mit
 a) $x \in B_{1,N}$, $gx = \overline{gb_1 x_{\overline{p}}}$, $\overline{s_1}^{-1} \overline{b}_1 x_{\overline{p}} \cap \overline{B}_{1,E} - \overline{b}_1 \overline{K} \neq \emptyset$
 impliziert $x = \ddot{u}[p,\overline{p}]$, $gb_1 x_p \neq \overline{gx}$ für alle $\overline{x} \in \overline{B}_1$

b) $t_1 s_1^{-1} \ddot{u}[p,\overline{p}] = \emptyset$.

Ist die Prämisse von 2a erfüllt, dann sei $\ddot{U}(p,\overline{p})$ der von $s_1^{-1}\ddot{u}[p,\overline{p}]$ erzeugte Untergraph von B_1 und $\overline{inc}:\ddot{U}(p,\overline{p}) \longrightarrow \overline{B}_1$

der durch $\overline{inc}\ddot{u}[p,\overline{p}] = \overline{b}_1 x_{\overline{p}}$

und $\overline{m}_1 \overline{inc} e = m_1 e$ für alle $e \in s_1^{-1}\ddot{u}[p,\overline{p}]$

eindeutig definierte Graphmorphismus. Wir erhalten das folgende Pushout in GRAPH:

$$\begin{array}{ccc} \ddot{U}(p,\overline{p}) & \xrightarrow{inc} & B_1 \\ {\scriptstyle \overline{inc}}\downarrow & & \downarrow{\scriptstyle \gamma} \\ \overline{B}_1 & \xrightarrow[\overline{\gamma}]{} & V(p,\overline{p}) \end{array} \tag{1}$$

(Wegen $t_1 s_1^{-1}\ddot{u}[p,\overline{p}] = \emptyset$ ist $V(p,\overline{p})$ wieder ein Σ-Graph.)

10 <u>Satz</u>

Sei P eine schwach unabhängige Σ-Grammatik. P ist schwach Church-Rosser bzgl. der Menge der Σ-Graphen, falls für alle Paare $(p,\overline{p}) \in P\times P$ - unter der Bedingung, daß es zwei direkte Ableitungen wie in Def.9 mit der Prämisse von 9.2a gibt - zwei Folgen

$$\{ a_i = (G_{i-1} \stackrel{p_i}{\Longrightarrow} G_i) \}_{1\leqslant i\leqslant n} \ , \ \{ a_i' = (G_{i-1}' \stackrel{p_i'}{\Longrightarrow} G_i') \}_{1\leqslant i\leqslant m}$$

direkter Ableitungen mit $G_0 = G_0' = V(p,\overline{p})$, $p_1 = p$, $p_1' = \overline{p}$, $p_i, p_i' \in P$ und $G_n = G_m'$ existieren derart, daß gilt:

1) Für alle $1\leqslant i\leqslant n$ ist die Wurzel von p_i in G_{i-1} residue von $\gamma b_1 x_p$, $\overline{\gamma}\overline{b}_1 x_{\overline{p}}$ oder eines in $V(p,\overline{p})$ "noch nicht vorhandenen" Knotens. Analog für p_i'.

2) Die Restabbildungen f_n und f_m' sind gleich (vgl. Kor.7).

Beweis:

Seien G Σ-Graph, $r,\overline{r} \in R_\Sigma$ sowie

$$G \stackrel{rp}{\Longrightarrow} H \quad \text{und} \quad G \stackrel{\overline{rp}}{\Longrightarrow} \overline{H} \tag{2}$$

zwei direkte Ableitungen.

p und $\overline{p}$ sind markierungskonsistent:

Für $x \varepsilon K$, $\overline{x} \varepsilon \overline{K}$ mit $m_1 b_1 x$, $\overline{m}_1 \overline{b}_1 \overline{x} \varepsilon C_{fix}$ und $gb_1 x = \overline{gb_1 \overline{x}}$ gilt wegen der schwachen Unabhängigkeit von P $x \neq x_p$ und $\overline{x} \neq x_{\overline{p}}$. Da p und $\overline{p}$ Σ-Produktionen sind, folgt $m_1 b_1 x = m_2 b_2 x$ und $\overline{m}_1 \overline{b}_1 \overline{x} = \overline{m}_2 \overline{b}_2 \overline{x}$ nach II.9.4b.

Fall 1: $gB_1 \cap \overline{gB_1} \subset gb_1 K \cap \overline{gb_1 \overline{K}}$.

Da die Ableitungen (2) natürlich sind (II.10), R_Σ vollständig ist (II.6) sowie $p, \overline{p}$ proper und markierungskonsistent sind, erhält man nach Satz 4 einen Graphen X, $s, \overline{s} \varepsilon R_\Sigma$ mit

$$H \xRightarrow{\overline{sp}} X \quad \text{und} \quad \overline{H} \xRightarrow{sp} X \text{ .}$$

Daraus folgt die Behauptung für p und $\overline{p}$.

Fall 2: Es gibt $x \varepsilon B_1$ und $\overline{x} \varepsilon \overline{B}_1 - \overline{b}_1 \overline{K}$ mit $gx = \overline{g\overline{x}}$.

$\overline{x}$ ist Kante, weil $\overline{p}$ fast ist, und es gilt $\overline{s}_1 \overline{x} = \overline{b}_1 x_{\overline{p}}$ wegen II.9.4c. Daraus folgt $gs_1 x = \overline{gb_1 x_{\overline{p}}}$, also $s_1 x = \ddot{u}[p,\overline{p}]$, da P schwach unabhängig ist. Man erhält

$$g\ddot{u}[p,\overline{p}] = \overline{gincu}[p,\overline{p}] \qquad (3)$$

nach Def. von $\overline{inc}$ (vgl. Def.9). Wir zeigen, daß

$$gy = \overline{gincy} \qquad \text{für alle } y \varepsilon \ddot{U}(p,\overline{p}) \qquad (4)$$

gilt. Sei e Kante in $\ddot{U}(p,\overline{p})$. Dann ist $s_1 e = \ddot{u}[p,\overline{p}]$, also $\overline{s}_1 \overline{ince} = \overline{incs_1 e} = \overline{b}_1 x_{\overline{p}}$ nach Def. von $\overline{inc}$. Daraus folgt $s_G ge = g\ddot{u}[p,\overline{p}] = \overline{gincu}[p,\overline{p}] = \overline{gb_1 x_{\overline{p}}} = \overline{gs_1 \overline{ince}} = s_G \overline{gince}$.

Außerdem gilt nach Def. von $\overline{inc}$ $m_G ge = m_1 e = \overline{m}_1 \overline{ince} = m_G \overline{gince}$, und wir erhalten $ge = \overline{gince}$, weil G Σ-Graph ist. Sei y Knoten in $\ddot{U}(p,\overline{p})$ mit $y \neq \ddot{u}[p,\overline{p}]$. Dann gibt es eine Kante e in $\ddot{U}(p,\overline{p})$ mit $s_1 e = \ddot{u}[p,\overline{p}]$ und $t_1 e = y$. Demnach gilt $gy = gt_1 e = t_G ge = t_G \overline{gince} = \overline{ginct_1 e} = \overline{gincy}$. Damit ist (4) gezeigt. Wegen der universellen Eigenschaft von (1) impliziert (4) einen Graphmorphismus $h: V(p,\overline{p}) \longrightarrow G$ mit $h\gamma = g$ und $h\overline{\gamma} = \overline{g}$. Ferner folgt aus (4):

$$rm_1 y = \overline{rm_1}\,\overline{incy} \qquad \text{für alle } y \in \ddot{U}(p,\overline{p}),$$

so daß - ebenfalls wegen der universellen Eigenschaft von (1) - eine eindeutige Markierungsfunktion m_V von $V(p,\overline{p})$ mit $m_V\gamma = rm_1$ und $m_V\overline{\gamma} = \overline{rm_1}$ existiert. Man erhält

$$m_G h\gamma = m_G g = rm_1 = m_V\gamma$$

und $$m_G h\overline{\gamma} = m_G g = rm_1 = m_V\overline{\gamma} ,$$

also $m_G h = m_V$, wieder weil (1) Pushout ist. Demnach ist $V(p,\overline{p})$ markierungsverträglich in G eingebettet, so daß die Behauptung des Satzes nach Voraussetzung aus Kor. 7 folgt, falls

a) der Rand von $V(p,\overline{p})$ in G keine kritischen Punkte enthält und

b) die Markierungen durch h identifizierter Randpunkte von allen direkten Ableitungen a_i und a_i' bewahrt werden.

Zu a): Da alle p_i und p_i' fast sind, bleiben alle Knoten von $V(p,\overline{p})$ durch sämtliche Ableitungen a_i und a_i' hindurch erhalten. Folglich gibt es keine kritischen Knoten. Wegen II.9.4c sind nur die outarcs der Wurzeln von p_i und p_i' in G_{i-1} bzw. G_{i-1}' kritische Kanten. Nach 10.1 sind solche Kanten residues der outarcs von $\gamma b_1 x_p$, $\overline{\gamma b_1} x_{\overline{p}}$ oder von in $V(p,\overline{p})$ noch nicht vorhandenen Knoten. Demnach sind die outarcs von $\gamma b_1 x_p$ und $\overline{\gamma b_1} x_{\overline{p}}$ die einzigen kritischen Kanten in $V(p,\overline{p})$. Zu jeder Randkante e in $V(p,\overline{p})$ gehört eine weitere Kante e' in $V(p,\overline{p})$ mit $e' \neq e$ und $he' = he$. Wäre e kritisch, dann wäre o.B.d.A. $s_V e = \gamma b_1 x_p$ und es würde einer der folgenden drei Fälle gelten:

a1) Es gibt $\overline{e} \in B_{1,E}$ mit $\overline{\gamma e} = e'$. Daraus folgt $gb_1 x_p =$
$h\gamma b_1 x_p = hs_V e = s_G he = s_G he' = hs_V e' = hs_V\overline{\gamma e} = h\overline{\gamma s_1}\,\overline{e}$

$= \overline{gs_1e}$, also ein Widerspruch zu 9.2a.

a2) $e' \in \gamma B_1$ und $s_V e' \neq \gamma b_1 x_p$. Damit erhält man analog zu a1) einen Widerspruch zu 9.1.

a3) $s_V e' = \gamma b_1 x_p$. Dann gilt $s_V e = s_V e'$. Wegen $m_V e = m_G he = m_G he' = m_V e'$ folgt $e = e'$, weil $V(p,\overline{p})$ ein Σ-Graph ist (vgl. Def.9). Wir hatten aber $e \neq e'$ vorausgesetzt.

Demnach gibt es keine kritischen Randkanten in $V(p,\overline{p})$.

Zu b): Sei $1 \leq i \leq n$. Da p_i Σ-Produktion ist, werden von a_i die Markierungen aller Knoten und Kanten von G_{i-1} außer der Wurzel von p_i bewahrt. Nach 10.1 kann dieser Knoten nur $\gamma b_1 x_p$ oder $\overline{\gamma b_1} x_{\overline{p}}$ als "Vorgänger" in $V(p,\overline{p})$ haben. Da P schwach unabhängig ist, gilt $x,y \notin \{\gamma b_1 x_p, \overline{\gamma b_1} x_{\overline{p}}\}$ für alle $x,y \in V(p,\overline{p})$ mit $x \neq y$ und $hx = hy$. Folglich erhält a_i die Markierungen aller durch h identifizierten Knoten und Kanten. Analog folgert man Bedingung b) für alle a_i' mit $1 \leq i \leq m$.

<u>Fall 3</u>: Es gibt $x \in B_1 - b_1 K$ und $\overline{x} \in \overline{B}_1$ mit $gx = \overline{gx}$.

Analog zu Fall 2 erhält man die Behauptung des Satzes.

11 <u>Definition</u>

Eine schwach unabhängige Σ-Grammatik P heißt STARK UNABHÄNGIG, wenn jede Produktion p in P einen Klebeknoten y_p besitzt derart, daß für alle $\overline{p} \in P$

$$y_p = \ddot{u}[p,\overline{p}]$$

gilt.

12 <u>Definition</u>

Sei P eine stark unabhängige Σ-Grammatik. $p \in P$ heißt ZEIGERUMSETZEND, falls gilt:

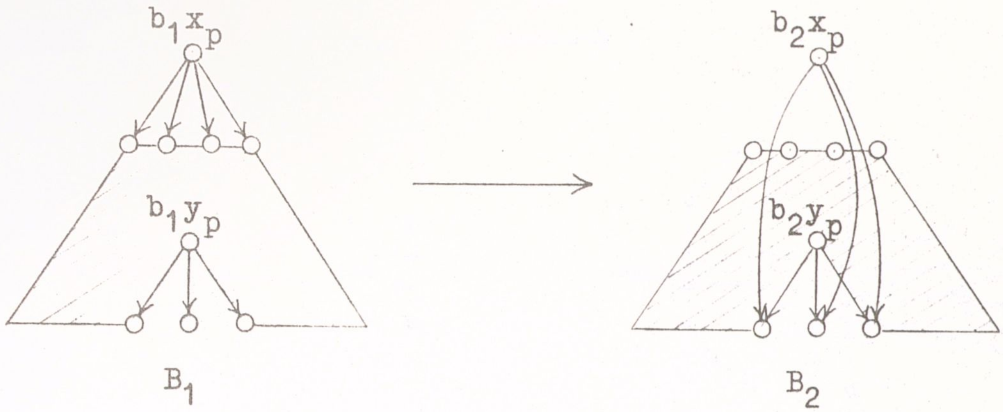
1) b_2 ist surjektiv auf Knoten

2) $e \in B_{2,E} - b_2 K \Longrightarrow s_2 e = b_2 x_p$

3) $m_1 b_1 y_p = m_2 b_2 x_p$

4) Sei $e_1 \in B_{1,E}$, $e_2 \in B_{2,E}$, $s_1 e_1 = b_1 y_p$, $s_2 e_2 = b_2 x_p$ und $m_1 e_1 = m_2 e_2$. $t_1 e_1 = b_1 x$ impliziert $t_2 e_2 = b_2 x$.

p läßt sich vereinfacht folgendermaßen darstellen:



13 <u>Definition</u>

Sei T eine Menge von Transformationsregeln (vgl. II.1) derart, daß $P = \{p(t) \mid t \in T\}$ stark unabhängig ist. T heißt ABGESCHLOSSEN, falls für alle

$$t = ((D,P,\mu),(D',P,\eta)) \in T$$

und $\sigma \in \Sigma$ mit $\text{Sorte}(\sigma) = \text{Typ}(y_{p(t)})$ und $\text{Rang}(\sigma) = n$

$$t' = ((D,P \cup \{x_1,\dots,x_n\},\mu'),(D',P \cup \{x_1,\dots,x_n\},\eta')) \in T$$

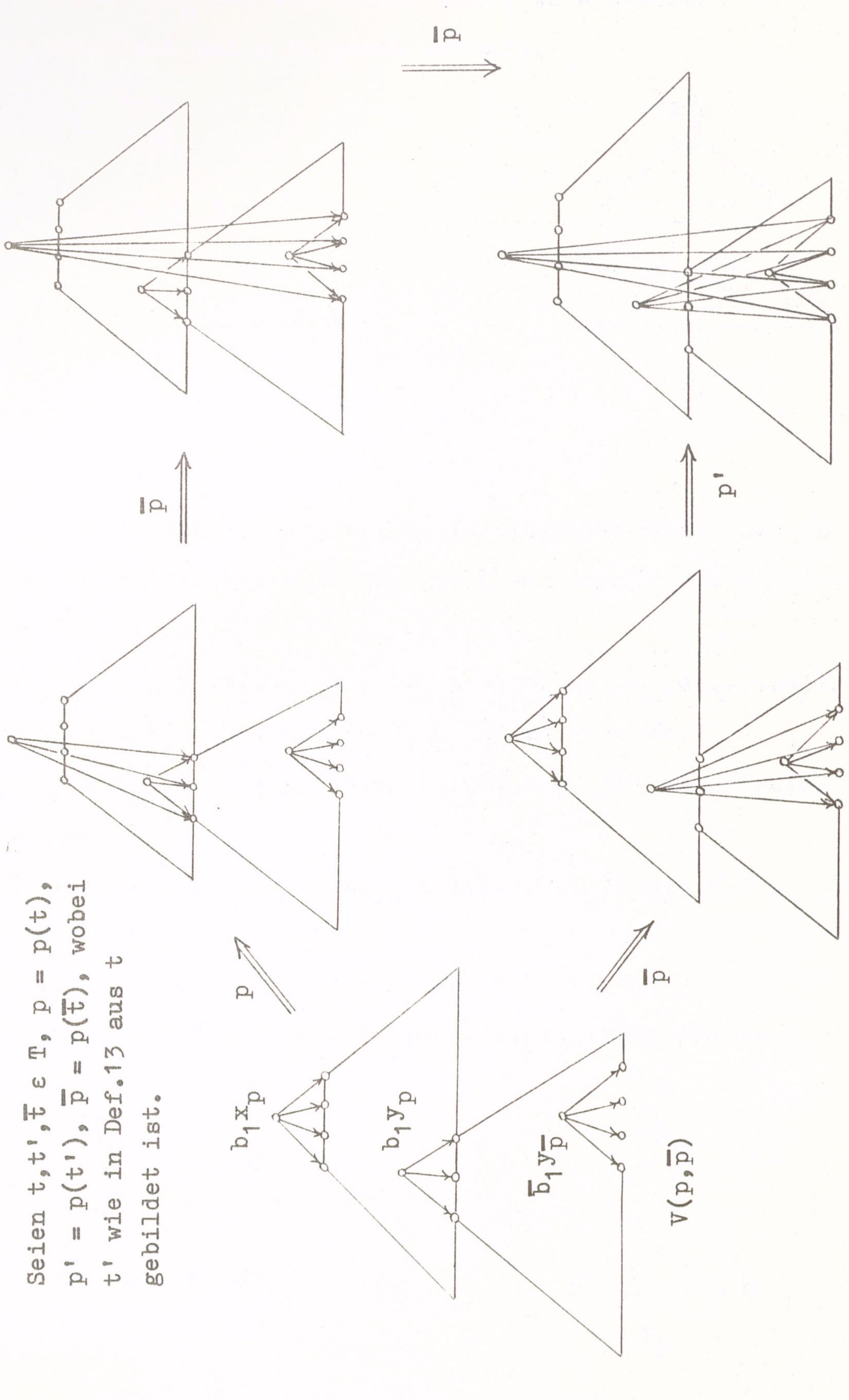
existiert mit $x_{p(t')} = x_{p(t)}$, $y_{p(t')} = y_{p(t)}$ und

$$\mu' x = \begin{cases} (\sigma,x_1,\dots,x_n) & \text{falls } x = y_{p(t')} \\ \mu x & \text{sonst} \end{cases}$$

14 <u>Satz</u>

Sei T eine abgeschlossene Menge von Transformationsregeln derart, daß $P = \{p(t) \mid t \in T\}$ stark unabhängig ist und nur aus zeigerumsetzenden Produktionen besteht. Dann ist P schwach Church-Rosser bzgl. Σ-Graphen.

Anschaulich erhält man die beiden Ableitungssequenzen auf der folgenden Seite, die den Bedingungen 10.1 und 10.2 genügen. Demnach folgt aus Satz 10 die Behauptung. Wir begnügen uns hier mit diesem anschaulichen Argument und verweisen auf die Einleitung, in der Gründe für den Verzicht auf eine formale Herleitung genannt sind.

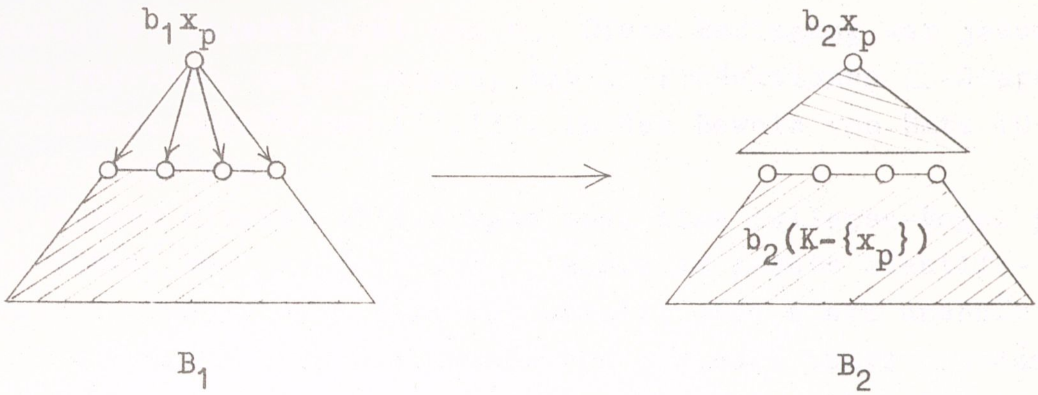


Seien $t, t', \bar{t} \in T$, $p = p(t)$, $p' = p(t')$, $\bar{p} = p(\bar{t})$, wobei t' wie in Def.13 aus t gebildet ist.

15 Definition

Eine Σ-Produktion p heißt UNZUSAMMENHÄNGEND, wenn $A = B_2 - b_2K \cup \{b_2x_p\}$ keinen Zyklus enthält und alle $x \in B_2$, die von b_2x_p aus erreichbar sind, zu A gehören.

Schematisch hat p folgende Gestalt:



16 Definition

Eine Graphproduktion p heißt COLLAPSE-REGEL, wenn B_1 und B_2 Σ-Graphen sind und $\sigma \in \Sigma$ mit Rang(σ)=n existiert, so daß gilt:

$B_{1,N} = \{x_p, y_p, x_1, \ldots, x_n\}$, $B_{1,E} = \{e_1, \ldots, e_n, e_1', \ldots, e_n'\}$,

$m_1x_p = m_1y_p = \sigma$, $m_1x_i \in C_{var}$ für alle $1 \leqslant i \leqslant n$,

$s_1e_i = x_p$, $s_1e_i' = y_p$, $t_1e_i = t_1e_i' = x_i$ für alle $1 \leqslant i \leqslant n$,

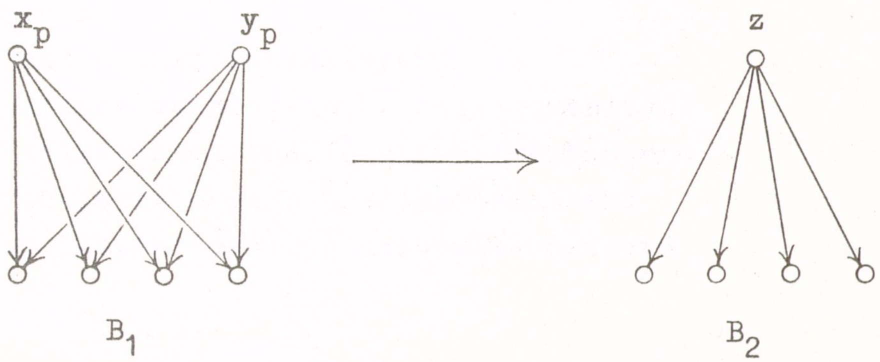
$K = B_1$,

$B_{2,N} = \{z, y_1, \ldots, y_n\}$, $B_{2,E} = \{f_1, \ldots, f_n\}$, $m_2z = \sigma$,

$b_2x_p = b_2y_p = z$,

$b_2x_i = y_i$, $b_2e_i = b_2e_i' = f_i$ für alle $1 \leqslant i \leqslant n$.

Daraus ergibt sich die folgende Darstellung von p:



17 Proposition

Die Menge der Σ-Graphen ist bzgl. Ableitungen über collapse-Regeln abgeschlossen.

collapse-Regeln haben alle Eigenschaften von Σ-Produktionen außer der Injektivität von b_2. Diese Bedingung war jedoch nur erforderlich, um zu zeigen, daß Σ-Produktionen Σ-Graphen in Σ-Graphen überführen (II.13). In den Beweis von Satz 10 geht sie nicht ein.

Wir wollen direkte Ableitungen über eine collapse-Regel p ausschließen, bei denen x_p und y_p schon im Ansatz identifiziert sind. Ohne diesen Fall ist nämlich jede stark unabhängige Σ-Grammatik nach Hinzunahme von p wieder stark unabhängig, und wir erhalten - vorbehaltlich eines formalen Beweises - den folgenden Church-Rosser-Satz.

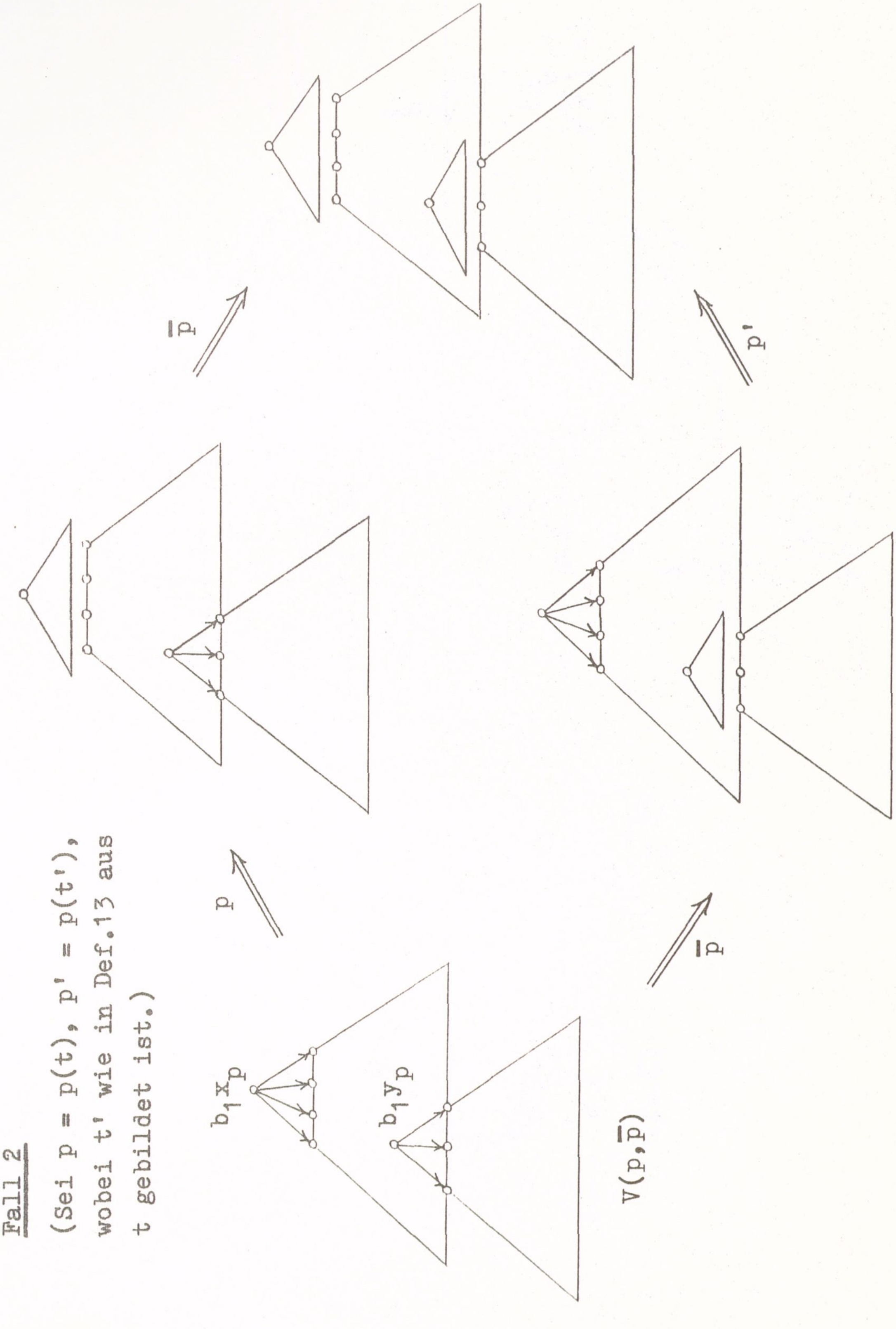
18 Satz

Sei T eine abgeschlossene Menge von Transformationsregeln derart, daß $P = \{p(t) \mid t \in T\}$ stark unabhängig ist und nur aus zeigerumsetzenden und unzusammenhängenden Produktionen besteht. Dann ist
$P' = P \cup \{\, p \text{ collapse-Regel} \mid \text{es gibt } \sigma \in \Sigma \text{ mit } m_1 x_p = \sigma \,\}$
schwach Church-Rosser bzgl. Σ-Graphen.

Für alle Paare $(p,\overline{p}) \in P' \times P'$ existieren (anschaulich) zwei in $V(p,\overline{p})$ beginnende konvergierende Ableitungssequenzen, welche die Voraussetzungen von Satz 10 erfüllen. Wir unterscheiden folgende Fälle:

1) p und $\overline{p}$ zeigerumsetzend
   Für diesen Fall ist die Behauptung durch Satz 14 gezeigt worden.
2) p und $\overline{p}$ unzusammenhängend
3) p unzusammenhängend, $\overline{p}$ zeigerumsetzend
4) p zeigerumsetzend, $\overline{p}$ unzusammenhängend
5) p collapse-Regel, $\overline{p}$ zeigerumsetzend
6) p collapse-Regel, $\overline{p}$ unzusammenhängend

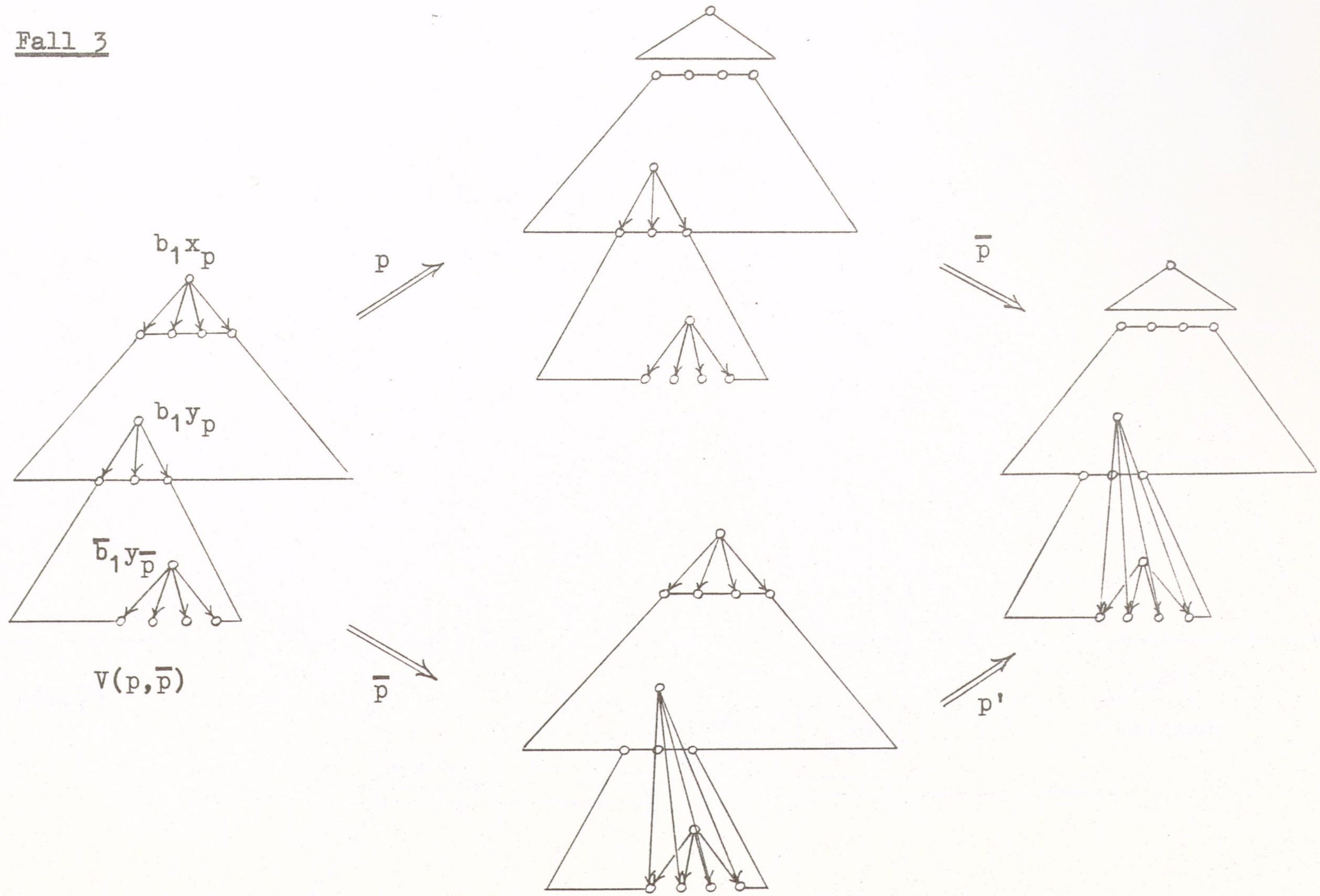
Im Fall " p zeigerumsetzend oder unzusammenhängend und $\overline{p}$ collapse-Regel " ist die Prämisse von 9.2a niemals erfüllt, weil $\overline{B}_1$ nur aus Klebepunkten besteht. Folglich sind nach Voraussetzung von Satz 10 keine Ableitungssequenzen für $V(p,\overline{p})$ erforderlich.



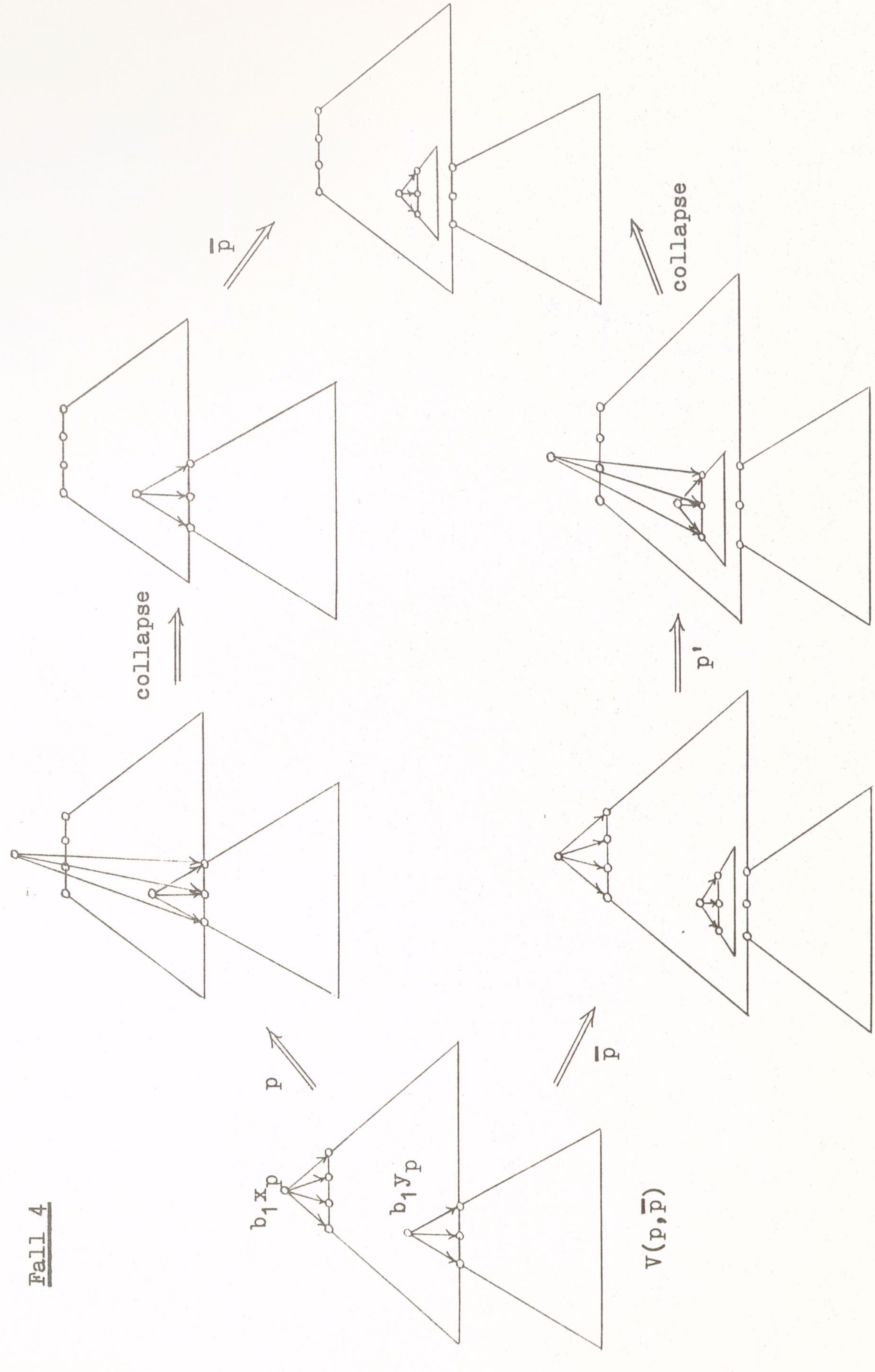
<u>Fall 2</u>

(Sei p = p(t), p' = p(t'), wobei t' wie in Def.13 aus t gebildet ist.)

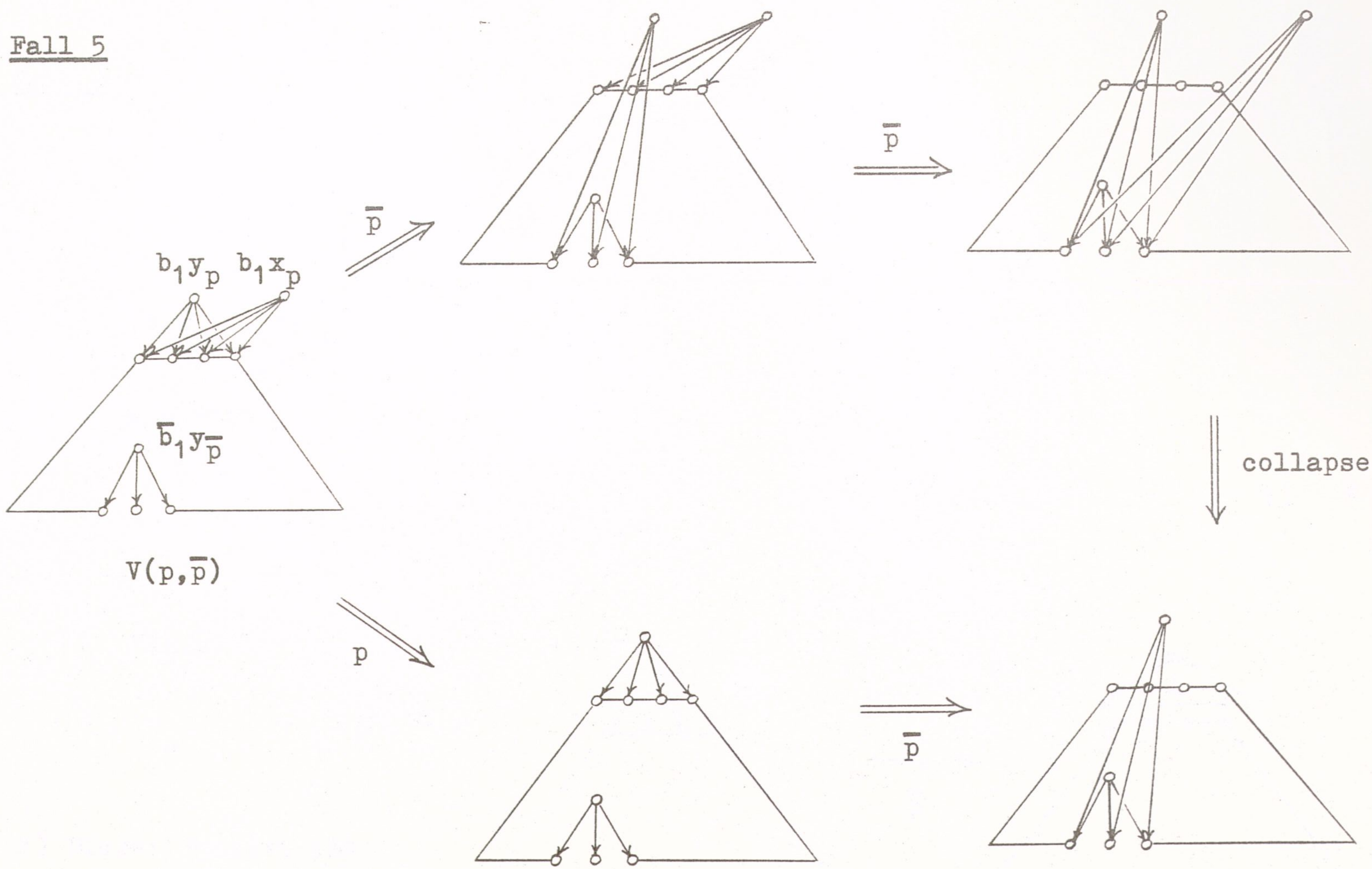
Fall 3



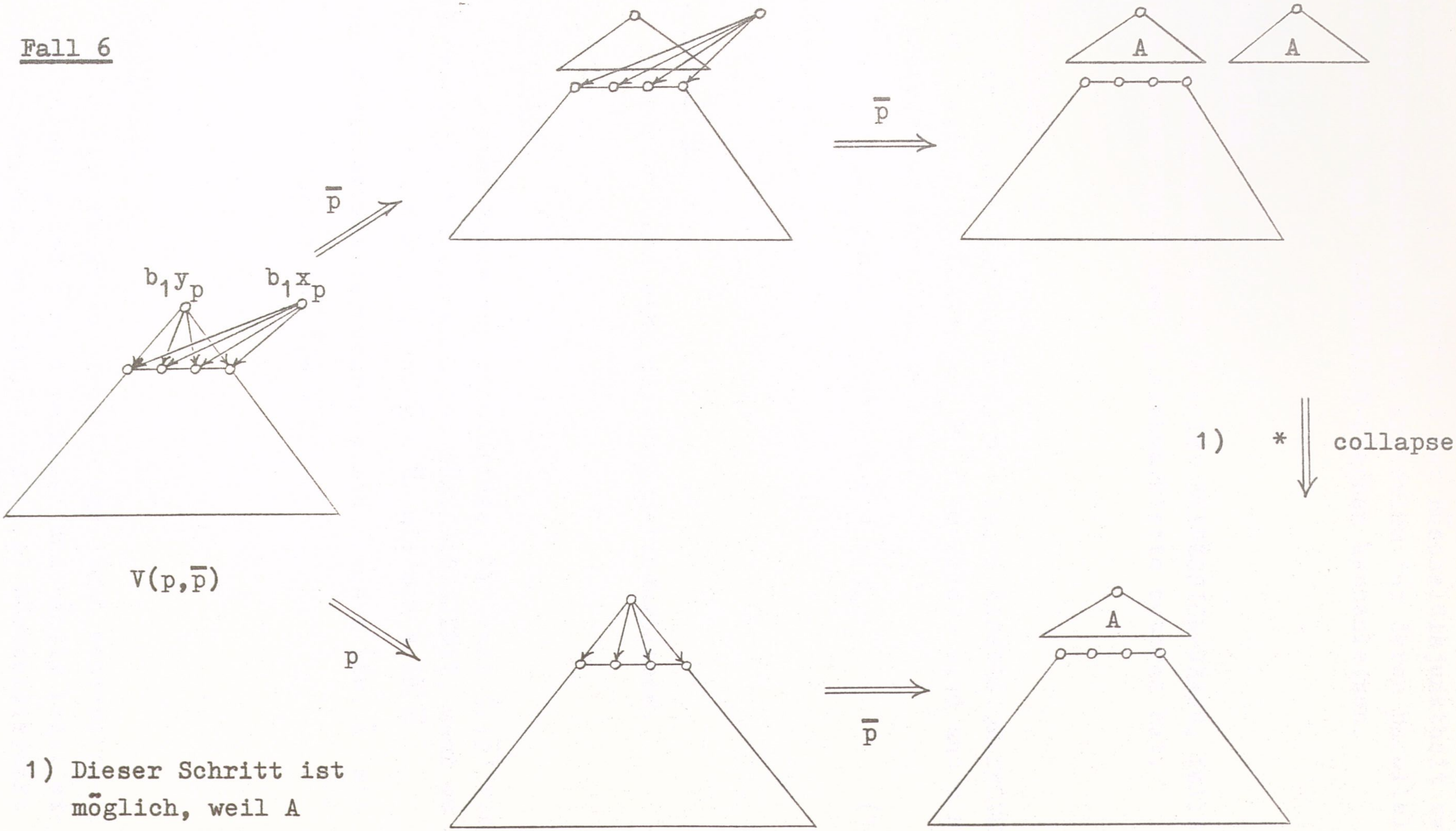
Fall 4


$b_1 x_p$
$b_1 y_p$
$V(p,\overline{p})$
p
$\overline{p}$
collapse
p'
$\overline{p}$
collapse

Fall 5



<u>Fall 6</u>



1) Dieser Schritt ist möglich, weil A azyklisch ist.

In Kapitel V wird Satz 18 auf LISP-Interpreter-Komponenten angewendet. Wir wollen jetzt mit der Büscheldisjunktheit zweier EXPRESSION-Produktionen ein Kriterium für deren parallele Unabhängigkeit bzgl. EXPRESSION-Graphen kennenlernen.

19 Definition

Ein markierter Graph G heißt EXPRESSION-GRAPH, falls kein Knoten von G zwei gleichmarkierte outarcs hat.

20 Definition

Eine propre Graphproduktion p heißt EXPRESSION-PRODUKTION, falls alle Kanten von B_1 fixmarkiert sind und ein fixmarkierter Knoten x_p in K mit

$$e \in B_{1,E} - b_1 K \Longrightarrow s_1 e = b_1 x_p \qquad (1)$$

existiert.

21 Proposition

Σ-Produktionen sind expression-Produktionen.

22 Definition

Sei R eine Menge von recolorings. Ein Paar $(p,\overline{p})$ von expression-Produktionen heißt BÜSCHELDISJUNKT, wenn es für alle Knoten $x \in B_1$ und $\overline{x} \in \overline{B}_1$ mit $m_1 x \in R\overline{m}_1 \overline{x}$ zwei gleichmarkierte outarcs e und $\overline{e}$ von x bzw. $\overline{x}$ mit

$$Rm_1 t_1 e \cap R\overline{m}_1 \overline{t}_1 \overline{e} = \emptyset$$

gibt.

23 Proposition

Sei R eine Menge von recolorings. Zwei expression-Produktionen p und $\overline{p}$ sind parallel unabhängig bzgl. expression-Graphen, wenn $(p,\overline{p})$ und $(\overline{p},p)$ büscheldisjunkt sind.

Beweis:

Sei G expression-Graph und $G \overset{rp}{\Longrightarrow} H$, $G \overset{\overline{rp}}{\Longrightarrow} \overline{H}$ zwei direkte Ableitungen mit $r, \overline{r} \in R$. Zu zeigen ist, daß alle $x \in B_1$ und $\overline{x} \in \overline{B}_1$ mit $gx = \overline{gx}$ Klebepunkte sind. Mit der Büscheldisjunktheit von $(p, \overline{p})$ weisen wir nach, daß x Klebepunkt ist. " $\overline{x}$ Klebepunkt " folgt dann analog aus der Büscheldisjunktheit von $(\overline{p}, p)$.

Da p proper ist, haben variabel markierte Punkte von B_1 Urbilder in K, so daß nur noch die beiden folgenden Fälle zu untersuchen sind:

1) x ist ein fixmarkierter Knoten. Dann gilt $m_1 x = rm_1 x = m_G gx = m_G \overline{gx} = \overline{rm}_1 \overline{x}$, und wir erhalten zwei gleichmarkierte outarcs e und $\overline{e}$ von x bzw. $\overline{x}$ mit

$$m_G t_G ge = rm_1 t_1 e \neq \overline{rm_1 t_1 e} = m_G t_G \overline{ge} \text{ ,} \qquad (2)$$

weil $(p, \overline{p})$ büscheldisjunkt ist. Da e und $\overline{e}$ fixmarkiert sind, folgt $m_G ge = m_G \overline{ge}$, und wegen $gx = \overline{gx}$ gilt $s_G ge = s_G \overline{ge}$, also $ge = \overline{ge}$, weil G expression-Graph ist. Das widerspricht jedoch (2).

2) x ist eine Kante. Falls $s_1 x \neq b_1 x_p$ gilt, dann ist x nach (1) Klebepunkt. $gx = \overline{gx}$ impliziert $gs_1 x = \overline{gs_1 x}$, und $b_1 x_p$ ist fixmarkiert, so daß man bei $s_1 x = b_1 x_p$ wie unter 1) einen Widerspruch erhält.

Satz 4 und Prop. 23 führen zu folgendem

24 <u>Satz</u>

Sei P eine Graph-Grammatik aus markierungskonsistenten expression-Produktionen und R eine vollständige Menge von recolorings. Sind alle Paare $(p, \overline{p}) \in P \times P$ büscheldisjunkt, dann ist P stark Church-Rosser bzgl. expression-Graphen.

Im letzten Teil dieses Kapitels wollen wir die in /ER 77/ begonnene Untersuchung der Vertauschbarkeit von Graphtransformationen mit garbage collection fortsetzen.

25 Definition

Ein markierter Graph G heißt ROOTED, falls G einen ausgezeichneten Knoten e_G besitzt. Es wird vorausgesetzt, daß in jeder direkten Ableitung

$$G \overset{p}{\Longrightarrow} H$$

e_G nicht zu $g(B_1-b_1K)$ gehört.

26 Definition

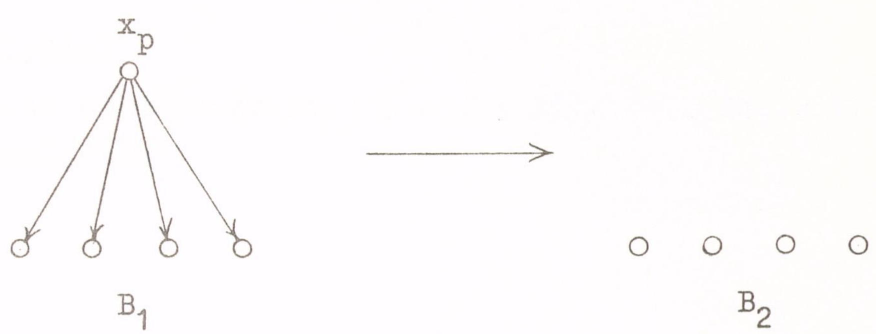
Eine Graphproduktion p heißt DELETE-REGEL, falls es ein $\sigma \varepsilon \Sigma$ mit Rang$(\sigma) = n$ gibt, so daß gilt:

$B_{1,N} = \{x_p, x_1, \ldots, x_n\}$, $B_{1,E} = \{e_1, \ldots, e_n\}$,

$m_1 x_p = \sigma$, $m_1 x_i \; \varepsilon \; C_{var}$, $s_1 e_i = x_p$, $t_1 e_i = x_i$ für alle $1 \leqslant i \leqslant n$,

$K = B_2 = \{x_1, \ldots, x_n\}$.

Demnach hat p folgende Gestalt:



27 Proposition

Die Menge der Σ-Graphen ist bzgl. Ableitungen über delete-Regeln abgeschlossen.

Die folgende Definition einer treelike Produktion ist (bis auf 28.1 für B_1, was zum Beweis von Satz 29 nicht gebraucht wird) eine geringfügige Abschwächung der entsprechenden Definition in /ER 77/.

28 Definition

Eine Graphproduktion p heißt TREELIKE, wenn es einen Knoten x_p in K gibt derart, daß folgende Bedingungen erfüllt sind:

1) Jeder Zyklus in B_i enthält mindestens einen Klebepunkt außer $b_i x_p$ $(i = 1,2)$.

2) $x \in B_{1,N} - \{b_1 x_p\} \Longrightarrow \text{indegree}(x) > 0$

3) $x \in K - \{x_p\} \Longrightarrow m_1 b_1 x = m_2 b_2 x$

4) $e \in B_{i,E} - b_i K,\ s_i e \neq b_i x_p \Longrightarrow s_i e \notin b_i K \quad (i = 1,2)$

29 Satz (/ER 77/,Th.6.4)

Sei P ein Church-Rosser-System bzgl. rooted Graphen, das aus treelike und fast Produktionen besteht. Dann ist auch $P' = P \cup \{\, p \text{ delete-Regel} \mid \text{es gibt } \sigma \in \Sigma \text{ mit } m_1 x_p = \sigma \}$ Church-Rosser bzgl. rooted Graphen.

30 Definition

Eine Graphproduktion p heißt INJEKTIV, falls b_1 und b_2 injektiv sind.

31 Satz

Sei R eine vollständige Menge von recolorings und P ein schwaches Church-Rosser-System bzgl. rooted Graphen, das aus Produktionen besteht, die treelike, injektiv und proper sind. Dann ist auch P' (vgl. Satz 29) schwach Church-Rosser bzgl. rooted Graphen.

Beweis:

Sei G rooted Graph, $p \in P$, $\overline{p}$ delete-Regel und $G \overset{rp}{\Longrightarrow} H$,

$G \overset{\overline{rp}}{\Longrightarrow} \overline{H}$ zwei direkte Ableitungen mit $r,\overline{r} \in R$.

<u>Fall 1</u>: $\overline{g}x_{\overline{p}} \notin gB_1$. (1)

Wir zeigen, daß p und $\overline{p}$ in diesem Fall parallel unabhängig sind. Für Kanten $e \in B_1$ und $\overline{e} \in \overline{B}_1$ mit $ge = \overline{ge}$ gilt $gs_1e = \overline{gs_1}\overline{e} = \overline{g}x_{\overline{p}}$, was (1) widerspricht. Folglich enthält $gB_1 \cap \overline{gB_1}$ keine Kanten. Knoten $\overline{x} \in \overline{B}_1$ mit $\overline{x} \neq x_{\overline{p}}$ sind Klebepunkte, und es gibt $\overline{e} \in \overline{B}_{1,E}$ mit $\overline{t}_1\overline{e} = \overline{x}$. Für $x \in B_1$ mit $gx = \overline{gx}$ gilt dann $t_G\overline{ge} = gx$. Wegen $\overline{ge} \in gB_1$ folgt daraus $gx \in gb_1K$ nach Klebebedingung II.12.2. Also ist unter Voraussetzung (1) $gB_1 \cap \overline{gB_1}$ in $gb_1K \cap \overline{gb_1}\overline{K}$ enthalten, d.h. p und $\overline{p}$ sind parallel unabhängig. p und $\overline{p}$ sind auch markierungskonsistent, weil $x \in K$ und $\overline{x} \in \overline{K}$ mit $gb_1x = \overline{gb_1}\overline{x}$ wegen (1) impliziert, daß $\overline{b}_1\overline{x}$ variabel markiert ist.

Damit erhalten wir nach Satz 4 zwei direkte Ableitungen

$$H \overset{\overline{sp}}{\Longrightarrow} X \quad \text{und} \quad \overline{H} \overset{sp}{\Longrightarrow} X$$

mit $s,\overline{s} \in R$ und X rooted nach /ER 77/,Lemma 3.3.

<u>Fall 2</u>: $\overline{g}x_{\overline{p}} \in gB_1$.

Nach 28.2 gibt es für alle $x \in B_{1,N}-\{b_1x_p\}$ $e \in B_{1,E}$ mit $t_Gge = gx$. Daraus folgt $\overline{g}x_{\overline{p}} \neq gx$, also $\overline{g}x_{\overline{p}} = gb_1x_p$. (2)

Wir zeigen:

a) $\overline{H} \underset{\text{delete}}{\overset{*}{\Longrightarrow}} \overline{X} = G-(g(B_1-b_1(K-\{x_p\})) \cup s_G^{-1}gb_1x_p)$

b) $H \underset{\text{delete}}{\overset{*}{\Longrightarrow}} X = H-(h(B_2-b_2(K-\{x_p\})) \cup s_H^{-1}hb_2x_p)$

c) $\overline{X} \cong X$

Zu a): Sei $A = \{z_1,\dots,z_n\}$ die Menge aller Knoten von $g(B_1-b_1(K-\{x_p\}))$. Die Indices der Elemente von A können so verteilt werden, daß gilt:

$z_1 = gb_1x_p$ und $s_Gt_G^{-1}z_{i+1} \subset \{z_1,\dots,z_i\}$ für alle $1 \leq i < n$

Dazu ist zu zeigen, daß Zyklen in G mindestens eine

ten aus G-A enthalten und daß für alle $e \in G_E$ $t_G e$

$s_G e \in A$ impliziert. Die erste Bedingung ist wegen 2

der Klebebedingung II.12.1 erfüllt. Sei $e \in G_E$. $t_G e$

ist ein Widerspruch zu (2). Zu zeigen bleibt also

$t_G e \in g(B_1-b_1K),\ s_G e \neq gb_1x_p \Longrightarrow s_G e \in g(B_1-b_1K)$

Aus der Prämisse von (3) folgt $e \in gB_1$ nach II.12.2

nach ist $s_G e \in gB_1$, und es existiert $e' \in B_1$ mit ge

Wäre $s_G e \in gb_1K$, dann gäbe es $x \in K$ mit $gs_1e' = s_G$

Daraus würde nach II.12.1 $s_1e' \in b_1K$ folgen, also e

nach 28.4 und schließlich $t_G e = t_G ge' = gt_1e' \in gb$

jedoch der Prämisse von (3) widerspricht. Damit ist

wiesen.

Durch Induktion über i zeigen wir, daß es für alle 0

Untergraphen $\overline{H}_i$ von G gibt mit

$$\overline{H}_{i,N} = G_N-\{z_1,\ldots,z_i\},\ \overline{H}_{i,E} = G_E-\bigcup_{j=1}^{i} s_G^{-1} z_j$$

und $\overline{H}_{i-1} \xrightarrow{\text{delete}} \overline{H}_i$.

Nach Voraussetzung gilt $\overline{H}_0 = G$, $\overline{H}_1 = H$ und $\overline{H}_0 \xrightarrow{\text{delete}}$

Sei (4) für i erfüllt. $e \in G_E$ mit $t_G e = z_{i+1}$ impliz

$s_G e \in s_G t_G^{-1} z_{i+1} \subset \{z_1,\ldots,z_i\}$, also $e \notin \overline{H}_{i,E}$. Außerd

$z_{i+1} \in g(B_1-b_1K)$ und daher $z_{i+1} \neq e_G$. Folglich ist de

auf z_{i+1} anwendbar, und man erhält

$$\overline{H}_i \xrightarrow{\text{delete}} \overline{H}_{i+1} \quad ,$$

also gilt (4) auch für i+1.

Zu zeigen bleibt $\overline{H}_{n,E} = \overline{X}_E$, d.h.

$s_G e \in \{z_1,\ldots,z_n\} \Longleftrightarrow e \in g(B_1-b_1K) \cup s_G^{-1} gb_1x_p$.

$s_G e \in g(B_1-b_1K)$ impliziert $e \in gB_1$ nach II.12.2 und

$e \notin gb_1K$, weil gb_1 Graphmorphismus ist. Sei $e \in g(B_1-b$

Daraus folgt $s_G e \in gB_1$, und es existiert $e' \in B_1$ mit $ge' = e$. Wäre $s_G e \in gb_1(K-\{x_p\})$, dann würde analog zum Beweis von (3) $e' \in b_1K$ gelten, und wir hätten mit $e = ge' \in gb_1K$ einen Widerspruch erhalten. Damit ist (5) gezeigt.

Zu b): Sei $A = \{z_1,\ldots,z_n\}$ die Menge aller Knoten von $h(B_2-b_2(K-\{x_p\}))$. Wir zeigen zunächst, daß ein Knoten z in A existiert, der keine einlaufenden Kanten hat.

Für alle $e \in B_{2,E}$ gilt

$$t_2e = b_2x_p \Longrightarrow e \notin b_2K \ . \qquad (6)$$

Wäre nämlich $t_2e = b_2x_p$ und $e \in b_2K$, dann gäbe es $e' \in D$ mit $c_2e' = he$. Daraus würde $c_2t_De' = t_Hc_2e' = t_Hhe = ht_2e = hb_2x_p = c_2dx_p$ folgen, also $t_De' = dx_p$, weil p injektiv ist, so daß $t_Gc_1e' = gb_1x_p$ gälte, was jedoch (2) widerspricht. (6) impliziert für alle $e \in B_{2,E}$

$$t_2e \in B_2-b_2(K-\{x_p\}) \Longrightarrow e \notin b_2K$$

und zusammen mit 28.4

$$t_2e \in B_2-b_2(K-\{x_p\}) \Longrightarrow s_2e \in B_2-b_2(K-\{x_p\}) \ . \qquad (7)$$

Daraus folgt, daß alle Vorgängerknoten von b_2x_p zu $B_2-b_2(K-\{x_p\})$ gehören. Wegen 28.1 hat einer dieser Knoten keine inarcs. Er heiße y. Ist $y = b_2x_p$, so hat auch hy keine inarcs, andernfalls gäbe es in $gb_1x_p = \overline{g}x_{\overline{p}}$ hineinlaufende Kanten. Ist $y \in B_2-b_2K$, so hat hy wegen der Klebebedingung II.12.2 keine inarcs. Demnach gibt es einen Knoten z in A ohne einlaufende Kanten.

Nach 28.1 und II.12.1 enthalten Zyklen in H mindestens einen Knoten aus H-A, und analog zu (7) erhält man $s_He \in A$ für alle $e \in H_E$ mit $t_He \in A$. Daher lassen sich die Elemente von A so indizieren, daß gilt:

$$z_1 = z \quad \text{und} \quad s_Ht_H^{-1}z_{i+1} \subset \{z_1,\ldots,z_i\} \text{ für alle } 1 \leqslant i < n$$

Analog zu (4) bildet man für alle $1 \leq i \leq n$ Untergraphen H_i von H mit $H_1 = H$, $H_{i-1} \xRightarrow{\text{delete}} H_i$ und $H_n = X$, wobei sich die letzte Gleichung aus der zu (5) analogen Äquivalenz

$$s_H e \in \{z_1, \ldots, z_n\} \Longleftrightarrow e \in h(B_2 - b_2 K) \cup s_H^{-1} h b_2 x_p$$

ergibt.

Zu c): Für alle $x \in K_N$ gilt

$$c_1 s_D^{-1} dx \subset s_G^{-1} c_1 dx$$

und, da c_1 injektiv ist,

$$c_1(D_E - s_D^{-1} dx) \subset c_1 D_E - s_G^{-1} c_1 dx \; .$$

Daraus folgt wegen der Injektivität von p

$$\begin{aligned}
\overline{X} &= G-(g(B_1-b_1(K-\{x_p\})) \cup s_G^{-1} g b_1 x_p) \\
&= G-(g((B_1-b_1K) \cup \{b_1 x_p\}) \cup s_G^{-1} g b_1 x_p) \\
&= G-g(B_1-b_1K)-\{g b_1 x_p\}-s_G^{-1} g b_1 x_p \\
&= c_1 D-\{c_1 d x_p\}-s_G^{-1} c_1 d x_p \\
&= c_1(D-\{d x_p\}-s_D^{-1} d x_p) \\
&\cong c_2(D-\{d x_p\}-s_D^{-1} d x_p) \\
&\;\;\vdots \\
&= H-(h(B_2-b_2(K-\{x_p\})) \cup s_H^{-1} h b_2 x_p) \\
&= X \; .
\end{aligned}$$

Kapitel V

DIE KORREKTHEIT UND VERTAUSCHBARKEIT VON TRANSFORMATIONS-REGELN EINES LISP-INTERPRETERS

Die in den Kapiteln I-IV entwickelte Theorie der Speicherzustände (Σ-Graphen) und Transformationsregeln (Σ-Produktionen) wird jetzt auf LISP-Ausdrücke bzw. LISP-Interpreter angewendet. Wir beschreiben die Syntax von LISP als einen viersortigen Operatorenbereich mit den Sorten atom (atoms), val (values), exp (expressions) und env (environments). Variablenfreie Ausdrücke sind i.a. vom Typ val. Terme vom Typ env sind Listen von Variablenbelegungen. Die LISP-Operation EVAL wertet einen exp-Term t aus, indem sie, einer solchen Belegungsliste entsprechend, die Variablen in t durch val-Terme ersetzt und so selbst einen val-Term abliefert.

Die Regeln zur Auswertung von LISP-Ausdrücken haben wir /HM 76/ entnommen und als Transformationsregeln im Sinne von Def. II.1 formuliert.

Da LISP-Ausdrücke einzelne Elemente (Atome), aber auch Listen oder Funktionen von Elementen eines Datenbereichs repräsentieren können, ist als Trägermenge der "semantischen" Algebra ein "rekursiver domain" (vgl. /Scott 70/) erforderlich. Wir werden die Existenz dieser Trägermenge aus dem erst kürzlich bewiesenen KATEGORIELLEN FIXPUNKTSATZ ableiten (vgl. /Ada 74/, /LS 77/, /Arb 77/).

Ausgehend von der Semantik der λ-Kalkül-Sprache SAL in /GTWW 77/, Kap.3.2, legen wir die Bedeutung der einzelnen LISP-Operationen fest und zeigen die Korrektheit o.g. Transformationsregeln bzgl. dieser Semantik mit Hilfe von Satz III.12.

Bei der Auswertung des LABEL-Operators in einem LISP-Ausdruck t wird ein Zyklus in der Graphrepräsentation von t erzeugt. Dieser Zyklus verändert sich bei weiteren Ableitungen jedoch nicht, und es gibt auch keine anderen Transformationen, bei denen Zyklen entstehen. Damit erhalten wir einerseits die schwache Unabhängigkeit der Σ-Grammatik unserer Transformationsregeln, andererseits können wir diese um delete-Regeln erweitern, mit denen nicht nur unreferenzierte Knoten, sondern auch nicht erreichbare Zyklen "eingesammelt" werden. Mit Hilfe der Sätze IV.18 und IV.29 läßt sich nachweisen, daß die Auswertungs- und delete-Regeln gemeinsam ein schwaches Church-Rosser-System bilden.

Leider haben wir damit noch nicht gezeigt, daß das Ergebnis der Auswertung unabhängig von der Reihenfolge ist, in der die Interpreter-Produktionen angewendet werden, daß also die Normalformen eines LISP-Ausdrucks nicht nur semantisch äquivalent, sondern gleich sind. In vielen Fällen erhält man mit einer Gewichtsfunktion weight auf Graphen aus der schwachen die allgemeine Church-Rosser-Eigenschaft (III.14) einer Graph-Grammatik (vgl. /ER 77/,5.3).

Die Anforderungen an weight, nämlich

$$\text{weight}(G) = 0 \text{ impliziert } G \text{ Normalform}$$

und

$$G \Longrightarrow H \text{ impliziert } \text{weight}(G) > \text{weight}(H),$$

werden jedoch von LISP-Ausdrücken bzw. -Interpretern nicht erfüllt, da sonst die Auswertung jedes Ausdrucks terminieren würde. Damit wäre das Halteproblem entscheidbar, weil es für jede partiell-rekursive Funktion f einen LISP-Ausdruck gibt, dessen funktionale Semantik f entspricht (vgl. /Kle 36/).

Da uns die allgemeine Church-Rosser-Eigenschaft aber nur zur Gewinnung eindeutiger Normalformen dient, besteht dennoch die Möglichkeit, daß durch Induktion über die Länge von Ableitungssequenzen (vgl. /Ros 73/,Kap.3, /ER 77/,5.5) aus der schwachen CR-Eigenschaft unserer Grammatik die Existenz einer operationellen Semantik für jeden LISP-Ausdruck hergeleitet werden kann.

1 Die Syntax von LISP

$S = \{atom, val, exp, env\}$ ist die Sortenmenge. Der Operatorenbereich Σ besteht aus

$a_i : \lambda \longrightarrow atom$ für alle $i \varepsilon \omega$
$true, false : \lambda \longrightarrow atom$
$quote : atom \longrightarrow val$
$car, cdr, is\text{-}atom : val \longrightarrow val$
$cons, eq, apply : val\ val \longrightarrow val$
$ite : val\ val\ val \longrightarrow val$
$X_i : \lambda \longrightarrow exp$ für alle $i \varepsilon \omega$
$QUOTE : atom \longrightarrow exp$
$CAR, CDR, IS\text{-}ATOM : exp \longrightarrow exp$
$LAMBDA\text{-}X_i, LABEL\text{-}X_i : exp \longrightarrow exp$ für alle $i \varepsilon \omega$
$CONS, EQ, APPLY : exp\ exp \longrightarrow exp$
$ITE : exp\ exp\ exp \longrightarrow exp$
$EVAL : exp\ env \longrightarrow val$
$NIL : \lambda \longrightarrow env$
$ATTR\text{-}X_i : val\ env \longrightarrow env$

Sei $Sp = (X,Y,Typ)$ ein S-sortiger Speicher. Die Speicherzustände von Sp bzgl. Σ heißen LISP-AUSDRÜCKE.

2 Ein Interpreter für LISP

Die folgende Menge T von Transformationsregeln (vgl.II.1) beschreibt die elementaren Auswertungsoperationen eines LISP-Interpreters. $t = ((D,P,\mu),(D',P',\eta))$ gehört zu T, falls II.1.1-3 und einer der folgenden Fälle für t gilt.

1) Es gibt $\sigma : s_1 \ldots s_n \longrightarrow val$ mit $t = t_{1,\sigma}$,

$D = D' = \{x_t, x_1, x_2\}$, $P = \{x_3, y_1, \ldots, y_n\}$,

$\mu x_t = (car, x_1)$, $\mu x_1 = (cons, x_2, x_3)$,

$\mu x_2 = (\sigma, y_1, \ldots, y_n)$, $\eta x_t = (\sigma, y_1, \ldots, y_n)$.

2) Es gibt $\sigma : s_1 \ldots s_n \longrightarrow exp$ mit $t = t_{2,\sigma}$,

$D = D' = \{x_t, x_1, x_2\}$, $P = \{x_3, y_1, \ldots, y_n\}$,

$\mu x_t = (CAR, x_1)$, $\mu x_1 = (CONS, x_2, x_3)$,

$\mu x_2 = (\sigma, y_1, \ldots, y_n)$, $\eta x_t = (\sigma, y_1, \ldots, y_n)$.

3) Es gibt $\sigma : s_1 \dots s_n \longrightarrow$ val mit $t = t_{3,\sigma}$,
$D = D' = \{x_t, x_1, x_3\}$, $P = \{x_2, y_1, \dots, y_n\}$,
$\mu x_t = (\mathrm{cdr}, x_1)$, $\mu x_1 = (\mathrm{cons}, x_2, x_3)$,
$\mu x_3 = \eta x_t = (\sigma, y_1, \dots, y_n)$.

4) Es gibt $\sigma : s_1 \dots s_n \longrightarrow$ exp mit $t = t_{4,\sigma}$,
$D = D' = \{x_t, x_1, x_3\}$, $P = \{x_2, y_1, \dots, y_n\}$,
$\mu x_t = (\mathrm{CDR}, x_1)$, $\mu x_1 = (\mathrm{CONS}, x_2, x_3)$,
$\mu x_3 = \eta x_t = (\sigma, y_1, \dots, y_n)$.

5) $t = t_5$, $D = \{x_t, x_1\}$, $P = \{x_2\}$, $D' = \{x_t, x_1, x_3\}$
$\mu x_t = (\text{is-atom}, x_1)$, $\mu x_1 = (\mathrm{quote}, x_2)$,
$\eta x_t = (\mathrm{quote}, x_3)$, $\eta x_3 = \mathrm{true}$.

6) $t = t_6$, $D = \{x_t, x_1\}$, $P = \{x_2\}$, $D' = \{x_t, x_1, x_3\}$
$\mu x_t = (\text{IS-ATOM}, x_1)$, $\mu x_1 = (\mathrm{QUOTE}, x_2)$,
$\eta x_t = (\mathrm{QUOTE}, x_3)$, $\eta x_3 = \mathrm{true}$.

7) $t = t_7$, $D = \{x_t, x_1\}$, $P = \{x_2, x_3\}$, $D' = \{x_t, x_1, x_4\}$
$\mu x_t = (\text{is-atom}, x_1)$, $\mu x_1 = (\mathrm{cons}, x_2, x_3)$,
$\eta x_t = (\mathrm{quote}, x_4)$, $\eta x_4 = \mathrm{false}$.

8) $t = t_8$, $D = \{x_t, x_1\}$, $P = \{x_2, x_3\}$, $D' = \{x_t, x_1, x_4\}$
$\mu x_t = (\text{IS-ATOM}, x_1)$, $\mu x_1 = (\mathrm{CONS}, x_2, x_3)$,
$\eta x_t = (\mathrm{QUOTE}, x_4)$, $\eta x_4 = \mathrm{false}$.

9) Es gibt $i \varepsilon \omega$ mit $t = t_{9,i}$,
$D = \{x_t, x_1, \dots, x_4\}$, $P = \emptyset$, $D' = D \cup \{x_5\}$,
$\mu x_t = (\mathrm{eq}, x_1, x_2)$, $\mu x_1 = (\mathrm{quote}, x_3)$, $\mu x_2 = (\mathrm{quote}, x_4)$,
$\mu x_3 = \mu x_4 = a_i$, $\eta x_t = (\mathrm{quote}, x_5)$, $\eta x_5 = \mathrm{true}$.

10) Es gibt $i \varepsilon \omega$ mit $t = t_{10,i}$,
$D = \{x_t, x_1, \dots, x_4\}$, $P = \emptyset$, $D' = D \cup \{x_5\}$,
$\mu x_t = (\mathrm{EQ}, x_1, x_2)$, $\mu x_1 = (\mathrm{QUOTE}, x_3)$, $\mu x_2 = (\mathrm{QUOTE}, x_4)$,
$\mu x_3 = \mu x_4 = a_i$, $\eta x_t = (\mathrm{QUOTE}, x_5)$, $\eta x_5 = \mathrm{true}$.

11) Es gibt $i,j \in \omega$ mit $a_i \neq a_j$, $t = t_{11,i,j}$,
$D = \{x_t, x_1, \ldots, x_4\}$, $P = \emptyset$, $D' = D \cup \{x_5\}$,
$\mu x_t = (\text{eq}, x_1, x_2)$, $\mu x_1 = (\text{quote}, x_3)$, $\mu x_2 = (\text{quote}, x_4)$,
$\mu x_3 = a_i$, $\mu x_4 = a_j$, $\eta x_t = (\text{quote}, x_5)$, $\eta x_5 = \text{false}$.

12) Es gibt $i,j \in \omega$ mit $a_i \neq a_j$, $t = t_{12,i,j}$,
$D = \{x_t, x_1, \ldots, x_4\}$, $P = \emptyset$, $D' = D \cup \{x_5\}$,
$\mu x_t = (\text{EQ}, x_1, x_2)$, $\mu x_1 = (\text{QUOTE}, x_3)$, $\mu x_2 = (\text{QUOTE}, x_4)$,
$\mu x_3 = a_i$, $\mu x_4 = a_j$, $\eta x_t = (\text{QUOTE}, x_5)$, $\eta x_5 = \text{false}$.

13) Es gibt $\sigma : s_1 \ldots s_n \longrightarrow \text{val}$ mit $t = t_{13,\sigma}$,
$D = D' = \{x_t, x_1, x_2, x_3\}$, $P = \{x_4, y_1, \ldots, y_n\}$,
$\mu x_t = (\text{ite}, x_1, x_3, x_4)$, $\mu x_1 = (\text{quote}, x_2)$, $\mu x_2 = \text{true}$,
$\mu x_3 = \eta x_t = (\sigma, y_1, \ldots, y_n)$.

14) Es gibt $\sigma : s_1 \ldots s_n \longrightarrow \text{exp}$ mit $t = t_{14,\sigma}$,
$D = D' = \{x_t, x_1, x_2, x_3\}$, $P = \{x_4, y_1, \ldots, y_n\}$,
$\mu x_t = (\text{ITE}, x_1, x_3, x_4)$, $\mu x_1 = (\text{QUOTE}, x_2)$, $\mu x_2 = \text{true}$,
$\mu x_3 = \eta x_t = (\sigma, y_1, \ldots, y_n)$.

15) Es gibt $\sigma : s_1 \ldots s_n \longrightarrow \text{val}$ mit $t = t_{15,\sigma}$,
$D = D' = \{x_t, x_1, x_2, x_4\}$, $P = \{x_3, y_1, \ldots, y_n\}$,
$\mu x_t = (\text{ite}, x_1, x_3, x_4)$, $\mu x_1 = (\text{quote}, x_2)$, $\mu x_2 = \text{false}$,
$\mu x_4 = \eta x_t = (\sigma, y_1, \ldots, y_n)$.

16) Es gibt $\sigma : s_1 \ldots s_n \longrightarrow \text{exp}$ mit $t = t_{16,\sigma}$,
$D = D' = \{x_t, x_1, x_2, x_4\}$, $P = \{x_3, y_1, \ldots, y_n\}$,
$\mu x_t = (\text{ITE}, x_1, x_3, x_4)$, $\mu x_1 = (\text{QUOTE}, x_2)$, $\mu x_2 = \text{false}$,
$\mu x_4 = \eta x_t = (\sigma, y_1, \ldots, y_n)$.

17) Es gibt $\sigma : s_1 \ldots s_n \longrightarrow \text{val}$ und $M = \{i_1, \ldots, i_k\} \subset \omega$ mit
$t = t_{17,\sigma,M}$, $D = D' = \{x_t, \text{var}, x_1, \ldots, x_k, z_k\}$,
$P = \{x_{k+1}, y_1, \ldots, y_n, z_1, \ldots, z_{k-1}\}$,
$\mu x_t = (\text{EVAL}, \text{var}, x_1)$, $\mu \text{var} = X_{i_k}$,
$\mu x_j = (\text{ATTR-}X_{i_j}, z_j, x_{j+1})$ für alle $1 \leqslant j \leqslant k$,

$\mu z_k = \eta x_t = (\sigma, y_1, \ldots, y_n)$.

18) $t = t_{18}$, $D = D' = \{x_t, x_1\}$, $P = \{x_2, x_3\}$,

$\mu x_t = (\mathrm{EVAL}, x_1, x_3)$, $\mu x_1 = (\mathrm{QUOTE}, x_2)$,

$\eta x_t = (\mathrm{quote}, x_2)$.

19) Es gibt $(\sigma, \sigma') \varepsilon \{(\mathrm{CAR},\mathrm{car}), (\mathrm{CDR},\mathrm{cdr}), (\mathrm{IS\text{-}ATOM},\mathrm{is\text{-}atom}), (\mathrm{CONS},\mathrm{cons}), (\mathrm{EQ},\mathrm{eq}), (\mathrm{APPLY},\mathrm{apply}), (\mathrm{ITE},\mathrm{ite})\}$ mit

$\mathrm{Rang}(\sigma) = n$, $t = t_{19,\sigma}$, $D = \{x_t, x_1\}$,

$P = \{x_2, y_1, \ldots, y_n\}$, $D' = \{x_t, x_1, z_1, \ldots, z_n\}$,

$\mu x_t = (\mathrm{EVAL}, x_1, x_2)$, $\mu x_1 = (\sigma, y_1, \ldots, y_n)$,

$\eta x_t = (\sigma', z_1, \ldots, z_n)$,

$\eta z_i = (\mathrm{EVAL}, y_i, x_2)$ für alle $1 \leqslant i \leqslant n$.

20) Es gibt $i \varepsilon \omega$ mit $t = t_{20,i}$,

$D = \{x_t, x_1, x_2\}$, $P = \{x_3, x_4, x_5\}$, $D' = D \cup \{x_6\}$,

$\mu x_t = (\mathrm{apply}, x_1, x_5)$, $\mu x_1 = (\mathrm{EVAL}, x_2, x_4)$,

$\mu x_2 = (\mathrm{LAMBDA\text{-}X}_i, x_3)$,

$\eta x_t = (\mathrm{EVAL}, x_3, x_6)$, $\eta x_6 = (\mathrm{ATTR\text{-}X}_i, x_5, x_4)$.

21) Es gibt $i, j \varepsilon \omega$ mit $X_i \neq X_j$, $t = t_{21,i,j}$,

$D = \{x_t, x_1, x_2\}$, $P = \{x_3, x_4\}$, $D' = D \cup \{x_5\}$,

$\mu x_t = (\mathrm{EVAL}, x_1, x_4)$, $\mu x_1 = (\mathrm{LABEL\text{-}X}_i, x_2)$,

$\mu x_2 = (\mathrm{LAMBDA\text{-}X}_j, x_3)$,

$\eta x_t = (\mathrm{EVAL}, x_2, x_5)$, $\eta x_5 = (\mathrm{ATTR\text{-}X}_i, x_t, x_4)$.

Obgleich wir uns bei der Formulierung der Transformationsregeln an /HM 76/ gehalten haben, sind ähnliche Regeln zur Auswertung von LISP-Ausdrücken z.B. auch in /Wand 76/,Kap.4 zu finden. Dort sind wie bei uns, aber im Gegensatz zu /HM 76/, Variablenbelegungen als "verkettete Listen" repräsentiert, und es gibt zusätzliche Regeln für die Suche nach einer bestimmten Variable in einer solchen Liste.

Die folgenden Ergebnisse aus /LS 77/ führen zur "semantischen" Algebra von LISP.

3 Definition

Eine Kategorie mit initialem Objekt und Colimiten von abzählbaren Morphismenketten (ω-Colimiten) heißt ω-KATEGORIE. Ein ω-FUNKTOR bewahrt ω-Colimiten.

4 Kategorieller Fixpunktsatz (/LS 77/,Th.1,2)

Sei K eine ω-Kategorie und $F:K \longrightarrow K$ ein ω-Funktor. Dann gibt es ein Objekt C in K und einen Isomorphismus $f:FC \longrightarrow C$ derart, daß für alle $D \in K$ und $g:FD \longrightarrow D$ eindeutig $h:C \longrightarrow D$ mit $hf = gFh$ existiert (f ist initiale F-Algebra). C heißt KLEINSTER FIXPUNKT von F.

5 Definition

Die Kategorie CPO^a (complete partial orders with adjunctions) hat als Objekte alle Halbordnungen mit Suprema von ω-Ketten und kleinstem Element und als Morphismen alle Paare (f,g) ω-stetiger Funktionen mit $gf = id$ und $fg \leqslant id$.

6 Satz (/LS 77/,Th.4)

CPO^a ist eine ω-Kategorie.

7 Definition

Sei $\{a_i\}_{i\in\omega}$ eine Menge, die wir als DATENBEREICH bezeichnen wollen. Auf $Atom = \{a_i\}_{i\in\omega} \cup \{true,false\} \dot{\cup} \{\perp\}$ ist durch

$$a \leqslant b \iff a = b \text{ oder } a = \perp$$

eine Halbordnung definiert, so daß Atom ein Objekt in CPO^a ist.

Sind $C,D \in CPO^a$, dann gehören auch $C+D = C \dot{\cup} D-\{\perp_C,\perp_D\}\dot{\cup}\{\perp\}$ mit kleinstem Element $\perp$, $C\times D$ mit komponentenweiser Halbordnung sowie die Menge $[C \longrightarrow D]$ der ω-stetigen Funktionen von C nach D zu CPO^a. Mit Hilfe von /LS 77/,Th.5 zeigt man, daß $+$, $\times$, $\longrightarrow$ und alle Kompositionen davon ω-Funktoren sind. Die Morphismenzuordnung durch $\longrightarrow$ macht die Beschränkung auf (Co-)Retraktionen in Def.5 erforderlich (vgl. /Arb 77/,S.219f.).

8 Die Semantik von LISP (vgl. 1 und I.9)

Trägermengen und Operationen der Σ-Algebra A sind wie folgt definiert:

A_{atom} = Atom (vgl. 7),

A_{val} ist der kleinste Fixpunkt des ω-Funktors $F: CPO^a \longrightarrow CPO^a$ mit $FC = Atom + C\times C + [C \longrightarrow C]$.

Sei $Var = \{X_i\}_{i\varepsilon\omega}$, $Val = A_{val}$. Dann induziert $\leqslant_{val}$ eine Halbordnung auf Val^{Var}, der Menge aller Abbildungen von Var nach Val, so daß auch

$A_{env} = Val^{Var}$ und

$A_{exp} = [Val^{Var} \longrightarrow Val]$ Objekte in CPO^a sind.

$a_{i,A} = a_i$ für alle $i \varepsilon \omega$,

$true_A = true$, $false_A = false$,

$quote_A = inc$,

$$car_A v = \begin{cases} v_1 & \text{falls } v \varepsilon Val\times Val \\ \perp & \text{sonst} \end{cases}$$

$$cdr_A v = \begin{cases} v_2 & \text{falls } v \varepsilon Val\times Val \\ \perp & \text{sonst} \end{cases}$$

$$is\text{-}atom_A v = \begin{cases} true & \text{falls } v \varepsilon Atom \\ false & \text{falls } v \varepsilon Val\times Val \\ \perp & \text{sonst} \end{cases}$$

$cons_A(v_1,v_2) = (v_1,v_2)$

$$eq_A(v_1,v_2) = \begin{cases} true & \text{falls } v_1,v_2 \varepsilon Atom \text{ und } v_1 = v_2 \\ false & \text{falls } v_1,v_2 \varepsilon Atom \text{ und } v_1 \neq v_2 \\ \perp & \text{sonst} \end{cases}$$

$$apply_A(v_1,v_2) = \begin{cases} v_1(v_2) & \text{falls } v_1 \varepsilon [Val \longrightarrow Val] \\ \perp & \text{sonst} \end{cases}$$

$$ite_A(v_1,v_2,v_3) = \begin{cases} v_2 & \text{falls } v_1 = \text{true} \\ v_3 & \text{falls } v_1 = \text{false} \\ \perp & \text{sonst} \end{cases}$$

$X_{i,A}(f) = f(X_i)$ für alle $i \in \omega$ und $f \in Val^{Var}$

$QUOTE_A(a)(f) = a$ für alle $f \in Val^{Var}$

$CAR_A(e) = car_A \circ e$,

$CDR_A(e) = cdr_A \circ e$,

$IS\text{-}ATOM_A(e) = is\text{-}atom_A \circ e$.

Die Funktion $assign\text{-}X_i : Val^{Var} \times Val \longrightarrow Val^{Var}$ verändert eine Variablenbelegung, indem sie X_i einen neuen Wert zuweist: $assign\text{-}X_i(f,v)(X_j) = \begin{cases} v & \text{falls } i = j \\ f(X_j) & \text{sonst} \end{cases}$

Sei $g : Val^{Var} \times Val \longrightarrow Val$. Dann ist

$$abstract(g) : Val^{Var} \longrightarrow [Val \longrightarrow Val]$$

definiert durch $abstract(g)(f)(v) = g(f,v)$.

$LAMBDA\text{-}X_{i,A}(e) = abstract(e \circ assign\text{-}X_i)$.

Die Funktion $lfp : [Val \longrightarrow Val] \longrightarrow Val$ liefert den kleinsten Fixpunkt einer ω-stetigen Abbildung auf Val (vgl. I.11).

$LABEL\text{-}X_{i,A}(e) = lfp \circ (LAMBDA\text{-}X_{i,A}(e))$.

Seien $e_1,\ldots,e_n \in A_{exp}$. Dann ist

$$[e_1,\ldots,e_n] : Val^{Var} \longrightarrow Val^n$$

definiert durch $[e_1,\ldots,e_n](f) = (e_1 f,\ldots,e_n f)$.

$CONS_A(e_1,e_2) = cons_A \circ [e_1,e_2]$,

$EQ_A(e_1,e_2) = eq_A \circ [e_1,e_2]$,

$APPLY_A(e_1,e_2) = apply_A \circ [e_1,e_2]$,

$ITE_A(e_1,e_2,e_3) = ite_A \circ [e_1,e_2,e_3]$,

$EVAL_A(e,f) = e(f)$,

$NIL_A = \perp$,

$ATTR\text{-}X_{i,A}(v,f) = assign\text{-}X_i(f,v)$.

Alle Operationen von A bewahren die Suprema von ω-Ketten (vgl. /Scott 72/,Kap.3, /Scott 74/,Kap.5). Demnach ist A eine ω-stetige Σ-Algebra.

9 <u>Satz</u>

Sei T die Menge der Transformationsregeln von 2. A ist eine Σ-$\mathcal{E}_T$-Algebra. Damit folgt aus Satz III.12, daß Ableitungen über T die funktionale Semantik von LISP-Ausdrücken bzgl. A erhalten, d.h. ein auf T basierender Interpreter arbeitet korrekt bzgl. A.

Beweis:

Sei $t = ((D,P,\mu),(D',P',\eta)) \in T$, $b \in A^Y$, $L = |\tilde{\mu}_{A(b)}|$ und $R = |\tilde{\eta}_{A(b)}|$. Dann ist $Lx_t = Rx_t$ zu zeigen (vgl. III.10(2)).

1) Sei $\sigma \in \Sigma$ mit $t = t_{1,\sigma}$ und $\text{Rang}(\sigma) = m$.

Daraus folgt $Lx_t = car_A Lx_1 = car_A cons_A(Lx_2,bx_3) =$
$car_A(Lx_2,bx_3) = Lx_2 = \sigma_A(by_1,\ldots,by_m) = Rx_t$ (vgl. I.14).

Analog zeigt man $Lx_t = Rx_t$ für alle

$$t \in T \cap \{t_{i,\sigma} \mid \sigma \in \Sigma,\ i \in \{3,13,15\}\}.$$

2) Sei $\sigma \in \Sigma$ mit $t = t_{2,\sigma}$ und $\text{Rang}(\sigma) = m$.

Daraus folgt $Lx_t = CAR_A Lx_1 = CAR_A CONS_A(Lx_2,bx_3) =$
$CAR_A(cons_A \circ [Lx_2,bx_3]) = car_A \circ cons_A \circ [Lx_2,bx_3] =$
$car_A \circ [Lx_2,bx_3] = Lx_2 = \sigma_A(by_1,\ldots,by_m) = Rx_t$.

Analog zeigt man $Lx_t = Rx_t$ für alle

$$t \in T \cap \{t_{i,\sigma} \mid \sigma \in \Sigma,\ i \in \{4,14,16\}\}.$$

3) Sei $t = t_5$.

Daraus folgt $Lx_t = \text{is-atom}_A Lx_1 = \text{is-atom}_A quote_A bx_2 =$
$true = true_A = quote_A true_A = quote_A Rx_3 = Rx_t$.

Analog zeigt man $Lx_t = Rx_t$ für alle

$$t \in \{t_7\} \cup \{t_{9,i} \mid i \in \omega\} \cup \{t_{11,i,j} \mid i,j \in \omega\}.$$

4) Sei $t = t_6$.

$Typ(x_2) = atom$ impliziert $bx_2 \varepsilon Atom$. Daher gilt für alle $f \varepsilon Val^{Var}$ $is\text{-}atom_A(QUOTE_A(bx_2)(f)) = is\text{-}atom_A bx_2 = true = true_A = QUOTE_A(true_A)(f)$.

Daraus folgt $Lx_t = IS\text{-}ATOM_A Lx_1 = IS\text{-}ATOM_A QUOTE_A bx_2 = is\text{-}atom_A \circ (QUOTE_A bx_2) = QUOTE_A true_A = QUOTE_A Rx_3 = Rx_t$.

Analog zeigt man $Lx_t = Rx_t$ für alle

$$t \varepsilon \{t_8\} \cup \{t_{10,i} \mid i \varepsilon \omega\} \cup \{t_{12,i,j} \mid i,j \varepsilon \omega\}.$$

5) Sei $\sigma \varepsilon \Sigma$ und $M = \{i_1,\ldots,i_k\} \subset \omega$ mit $t = t_{17,\sigma,M}$ und $Rang(\sigma) = m$.

Wir zeigen für alle $0 \leqslant j < k$ durch Induktion über j:

$$Lx_{k-j}(X_{i_k}) = Lx_k(X_{i_k}) \qquad (1)$$

$j = 0$ ist klar. Sei (1) für j erfüllt. Dann gilt

$Lx_{k-(j+1)}(X_{i_k}) = ATTR\text{-}X_{i_{k-(j+1)},A}(bz_{k-(j+1)}, Lx_{k-j})(X_{i_k})$

$= assign\text{-}X_{i_{k-(j+1)}}(Lx_{k-j}, bz_{k-(j+1)})(X_{i_k}) = Lx_{k-j}(X_{i_k})$.

Also ist (1) auch für j+1 erfüllt.

Daraus folgt $Lx_t = EVAL_A(Lvar, Lx_1) = EVAL_A(X_{i_k,A}, Lx_1)$

$= X_{i_k,A}(Lx_1) = Lx_1(X_{i_k}) = Lx_k(X_{i_k})$

$= ATTR\text{-}X_{i_k,A}(Lz_k, bx_{k+1})(X_{i_k})$

$= assign\text{-}X_{i_k}(bx_{k+1}, Lz_k)(X_{i_k}) = Lz_k = \sigma_A(by_1,\ldots,by_m)$

$= Rx_t$.

6) Sei $t = t_{18}$.

Daraus folgt $Lx_t = EVAL_A(Lx_1, bx_3) = EVAL_A(QUOTE_A bx_2, bx_3) = QUOTE_A(bx_2)(bx_3) = bx_2 = quote_A bx_2 = Rx_t$.

7) Sei $\sigma \varepsilon \Sigma$ mit $t = t_{19,\sigma}$ und $Rang(\sigma) = m$.

Dann gilt $Lx_t = EVAL_A(Lx_1, bx_2) = Lx_1(bx_2) =$

$\sigma_A(by_1,\ldots,by_m)(bx_2) = \sigma'_A\,[by_1,\ldots,by_m](bx_2) =$

$\sigma'_A((by_1)(bx_2),\ldots,(by_m)(bx_2)) =$

$\sigma'_A(EVAL_A(by_1,bx_2),\ldots,EVAL_A(by_m,bx_2)) =$

$\sigma'_A(Rz_1,\ldots,Rz_m) = Rx_t$.

8) Sei $i \in \omega$ mit $t = t_{20,i}$.

Daraus folgt $Lx_t = apply_A(Lx_1,bx_5) =$

$apply_A(EVAL_A(Lx_2,bx_4),bx_5) =$

$apply_A(EVAL_A(LAMBDA\text{-}X_{i,A}(bx_3),bx_4),bx_5) =$

$apply_A(EVAL_A(abstract((bx_3)\circ assign\text{-}X_i),bx_4),bx_5) =$

$apply_A(abstract((bx_3)\circ assign\text{-}X_i)(bx_4),bx_5) =$

$abstract((bx_3)\circ assign\text{-}X_i)(bx_4)(bx_5) =$

$(bx_3)\circ assign\text{-}X_i(bx_4,bx_5) = (bx_3)\circ ATTR\text{-}X_{i,A}(bx_5,bx_4) =$

$(bx_3)(Rx_6) = EVAL_A(bx_3,Rx_6) = Rx_t$.

9) Seien $i,j \in \omega$ mit $t = t_{21,i,j}$.

Wir zeigen zunächst

$$Lx_t = lfp(abstract((Lx_2)\circ assign\text{-}X_i)(bx_4)) . \tag{2}$$

$Lx_t = EVAL_A(Lx_1,bx_4) = EVAL_A(LABEL\text{-}X_{i,A}(Lx_2),bx_4)$

$= EVAL_A(lfp\circ(LAMBDA\text{-}X_{i,A}(Lx_2)),bx_4)$

$= EVAL_A(lfp\circ(abstract((Lx_2)\circ assign\text{-}X_i)),bx_4)$

$= lfp\circ(abstract((Lx_2)\circ assign\text{-}X_i))(bx_4)$.

Aus (2) erhält man für alle $n \in \omega$

$$\tilde{\eta}^{\,n}_{A(b)}(\perp)x_t \leqslant Lx_t \tag{3}$$

durch Induktion über n:

$n = 0$ ist klar. $\tilde{\eta}_{A(b)}(\perp)x_t = EVAL_A(\perp x_2, \perp x_5) =$

$= \mathrm{EVAL}_A(\perp,\perp) = \perp(\perp) = \perp \leqslant Lx_t$.

Sei (3) für n erfüllt. Dann gilt

$\tilde{\eta}^{\,n+2}_{A(b)}(\perp)x_t = \mathrm{EVAL}_A(\tilde{\eta}^{\,n+1}_{A(b)}(\perp)x_2, \tilde{\eta}^{\,n+1}_{A(b)}(\perp)x_5) =$

$\mathrm{EVAL}_A(\text{LAMBDA-X}_{j,A}(bx_3), \text{ATTR-X}_{i,A}(\tilde{\eta}^{\,n}_{A(b)}(\perp)x_t, bx_4)) =$

$\mathrm{EVAL}_A(Lx_2, \text{ATTR-X}_{i,A}(\tilde{\eta}^{\,n}_{A(b)}(\perp)x_t, bx_4)) \leqslant$

$\mathrm{EVAL}_A(Lx_2, \text{ATTR-X}_{i,A}(Lx_t, bx_4)) =$

$\mathrm{EVAL}_A(Lx_2, \text{assign-X}_i(bx_4, Lx_t)) =$

$(Lx_2)(\text{assign-X}_i(bx_4, Lx_t)) =$

$\text{abstract}((Lx_2)\circ\text{assign-X}_i)(bx_4)(Lx_t) \overset{(2)}{=} Lx_t$.

Also ist (3) auch für n+2 erfüllt.

Schließlich gilt

$$Rx_t = \text{abstract}((Lx_2)\circ\text{assign-X}_i)(bx_4)(Rx_t) . \tag{4}$$

$Rx_t = \mathrm{EVAL}_A(Rx_2, Rx_5) =$

$\mathrm{EVAL}_A(\text{LAMBDA-X}_{j,A}(bx_3), \text{ATTR-X}_{i,A}(Rx_t, bx_4)) =$

$\mathrm{EVAL}_A(Lx_2, \text{assign-X}_i(bx_4, Rx_t)) =$

$(Lx_2)(\text{assign-X}_i(bx_4, Rx_t)) =$

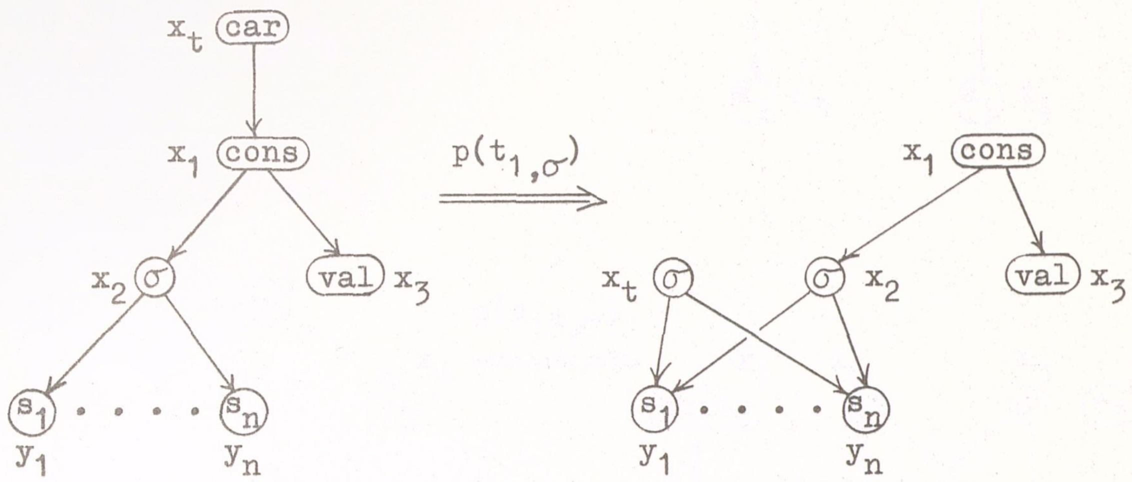
$\text{abstract}((Lx_2)\circ\text{assign-X}_i)(bx_4)(Rx_t)$.

(2) und (4) implizieren $Lx_t \leqslant Rx_t$, während aus (3) $Rx_t \leqslant Lx_t$ folgt.

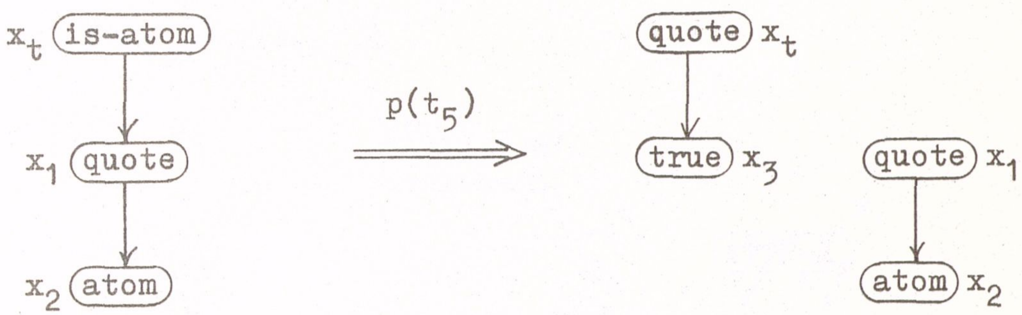
Bevor wir uns der schwachen Church-Rosser-Eigenschaft von T zuwenden, stellen wir zunächst die Graphrepräsentationen der einzelnen Transformationsregeln anschaulich dar (vgl. II.18).

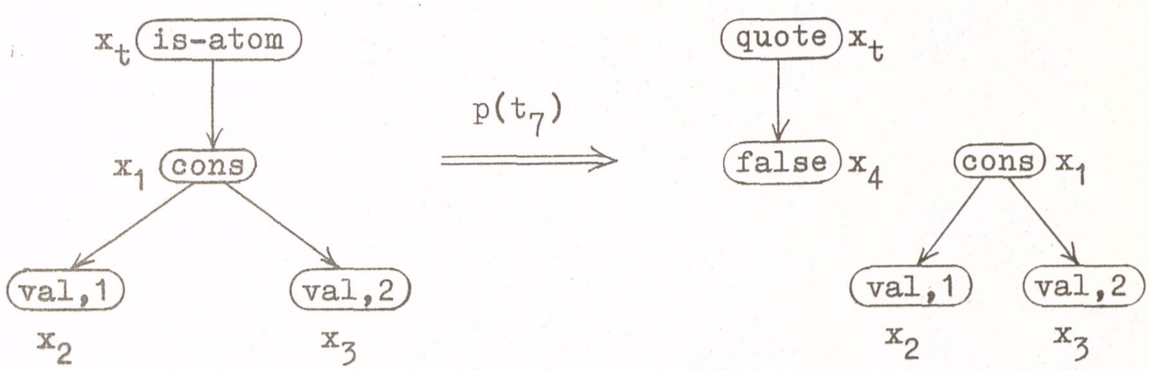
10 Der LISP-Interpreter als Graph-Grammatik

Sei T die Menge der Transformationsregeln von 2. Nach Prop. II.18 erhalten wir aus T die folgende Σ-Grammatik $P = \{p(t) \mid t \in T\}$.

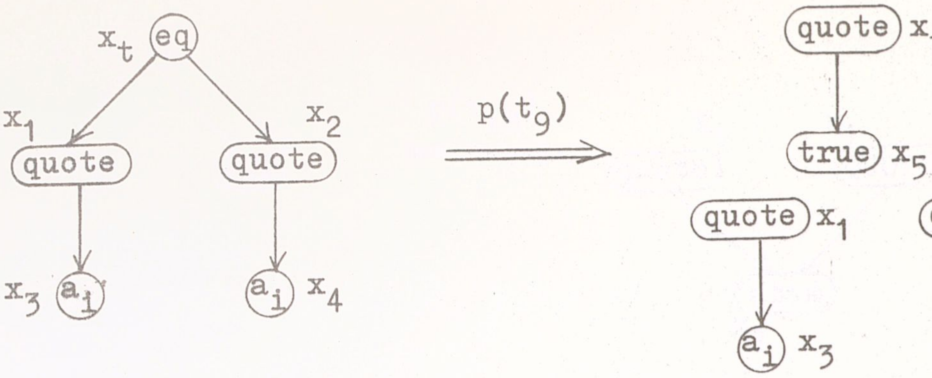


Ähnlich sehen die Darstellungen von $p(t_{i,\sigma})$ mit $i \in \{2,3,4\}$ aus.

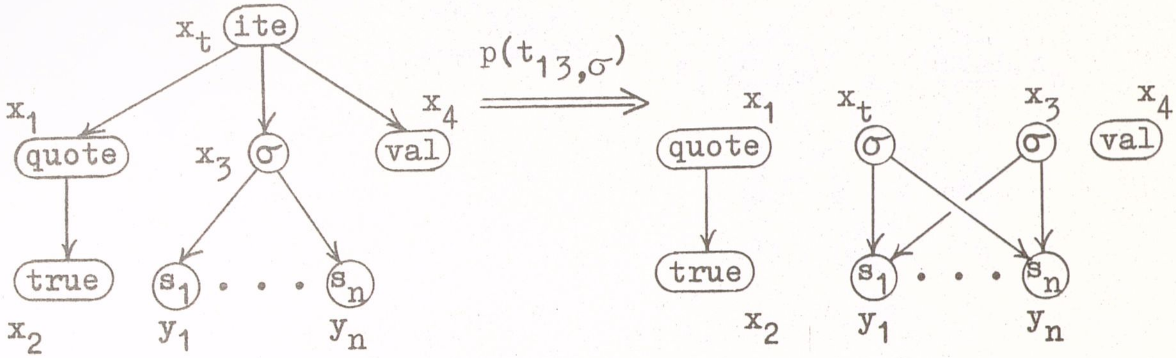




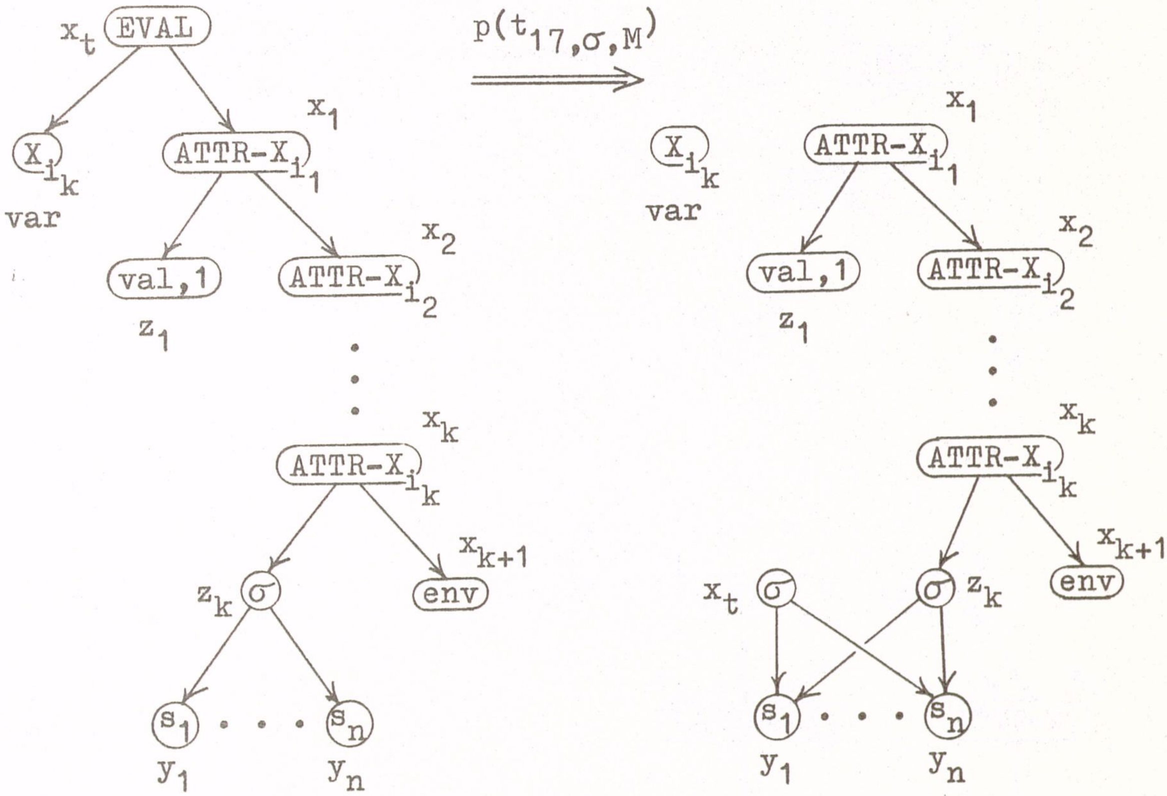
Analoge Darstellungen erhält man für $p(t_6)$ und $p(t_8)$.

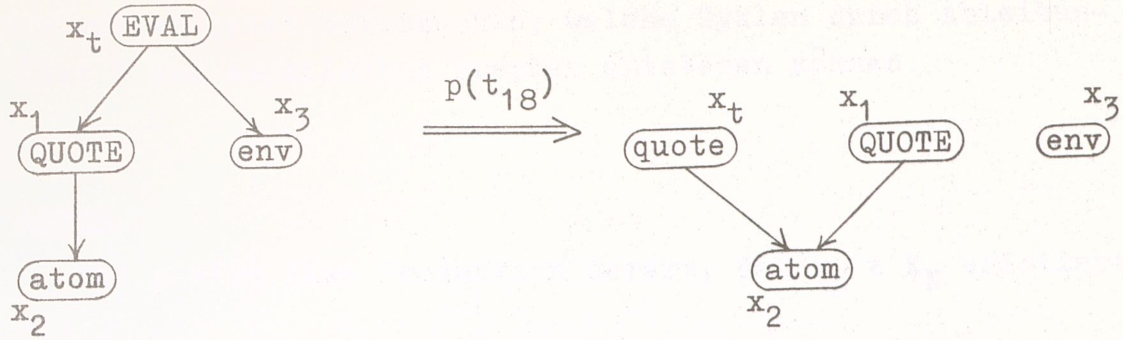


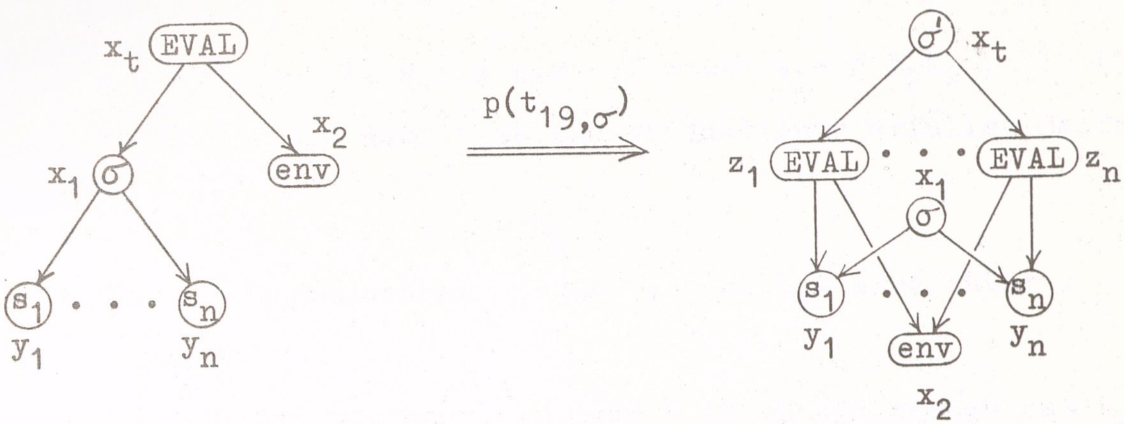
Analog: $p(t_{10})-p(t_{12})$.

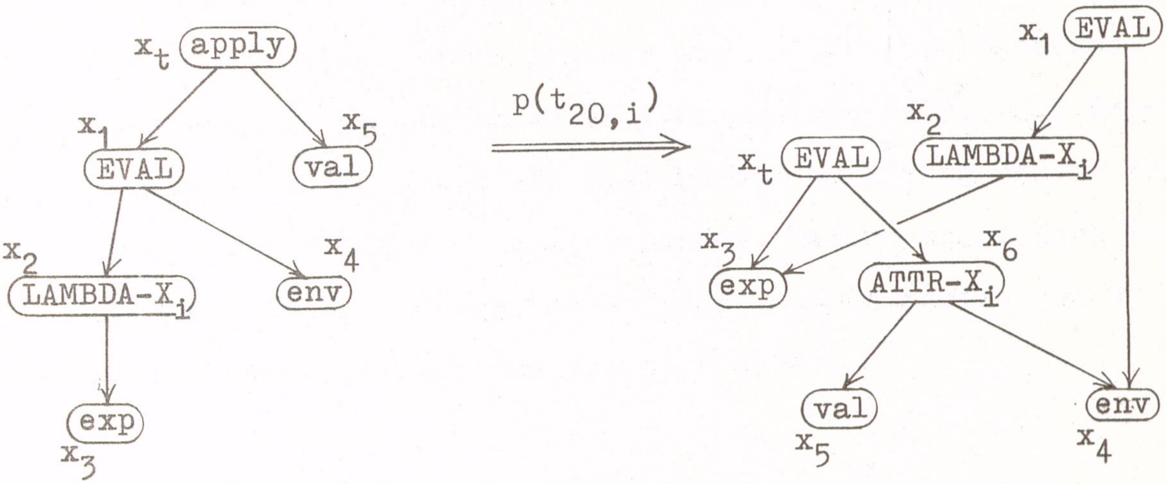


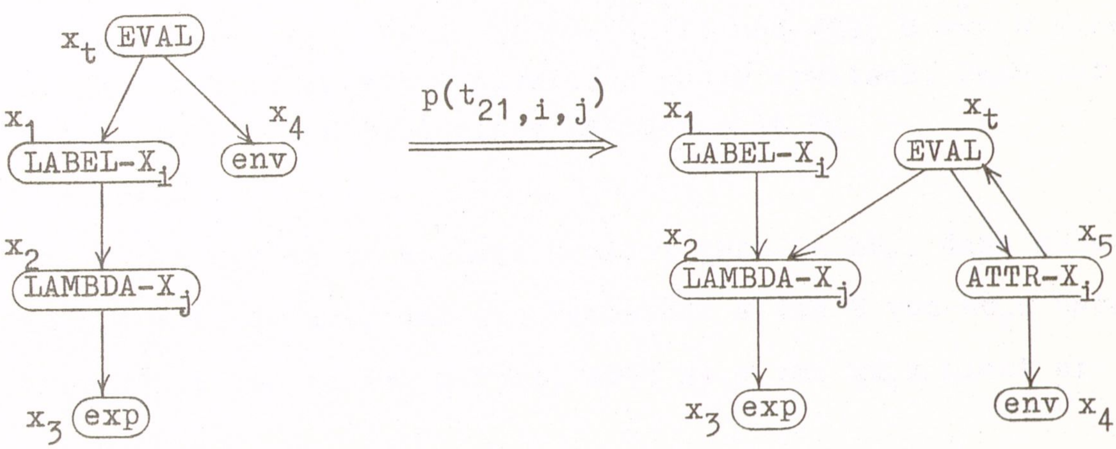
Analog: $p(t_{14,\sigma})-p(t_{16,\sigma})$.




EVAL
QUOTE
env
atom
p(t18)
quote
QUOTE
env
atom


EVAL
env
p(t19,σ)
EVAL
EVAL
env


apply
EVAL
val
LAMBDA-Xi
env
exp
p(t20,i)
EVAL
EVAL
LAMBDA-Xi
exp
ATTR-Xi
val
env


EVAL
LABEL-Xi
env
LAMBDA-Xj
exp
p(t21,i,j)
LABEL-Xi
EVAL
LAMBDA-Xj
ATTR-Xi
exp
env

Wir wollen jetzt untersuchen, welche Zyklen durch Ableitungen über P aus azyklischen Graphen entstehen können.

11 Lemma

Sei p eine fast Produktion derart, daß $x_p \in K_N$ existiert mit

$$b_1 x \text{ ist von } b_1 x_p \text{ erreichbar für alle } x \in K_N \qquad (1)$$

und

$$e \in B_{2,E} - b_2 K,\ s_2 e \neq b_2 x_p \Longrightarrow s_2 e \notin b_2 K_N\ . \qquad (2)$$

(Sowohl Σ- als auch treelike Produktionen erfüllen Bedingung (2).)

Dann gilt für alle $x,y \in K_N$

$b_2 y$ von $b_2 x$ erreichbar $\Longrightarrow$ $b_1 y$ von $b_1 x$ erreichbar .

Beweis:

Jeder Weg von $b_2 x$ nach $b_2 y$ läßt sich in Kantenzüge aus $b_2 K_E$ und solche aus $B_{2,E} - b_2 K$ (mit Anfangs- und Endknoten aus $b_2 K_N$) aufteilen. Wege, die nur aus Klebekanten bestehen, verbinden in B_1 die "gleichen" Knoten wie in B_2. Ein Kantenzug w aus $B_{2,E} - b_2 K$ mit Klebeknoten am Anfang und am Ende kann wegen (2) nur in $b_2 x_p$ beginnen. Daraus folgt nach (1), daß auch in B_1 ein Weg vom Anfangs- zum Endknoten von w führt. Daher ist $b_1 y$ von $b_1 x$ erreichbar.

12 Lemma

Sei p eine fast Produktion mit (1) und (2), $G \overset{p}{\Longrightarrow} H$ eine direkte Ableitung mit G azyklisch und H zyklisch. Dann ist jeder Zyklus in H Bild eines Zyklusses in B_2.

Beweis:

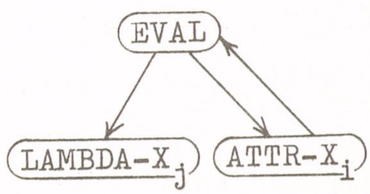
Sei Z ein Zyklus in H. Läge Z nicht ganz in hB_2, dann gäbe es $x,y \in K_N$ derart, daß ein Teilstück w von Z von $hb_2 x$ nach $hb_2 y$ führt und zu hB_2 gehört, aber $gb_1 y$ von $gb_1 x$ nicht erreichbar ist. Die erste Bedingung an x,y impliziert jedoch, daß $h^{-1}w$ aus einer Folge $(w_1,\ldots,w_n)$ unzusammenhängender Kantenzüge besteht, wobei h für alle $1\leq i<n$ den Endknoten y_i von w_i mit dem Anfangsknoten x_{i+1} von w_{i+1} identifiziert, b_2x der Anfangsknoten x_1 von w_1 und b_2y der Endknoten y_n von w_n ist. Nach Klebebedingung II.12.1 sind x_i,y_i für alle $1\leq i\leq n$ Klebepunkte, also gibt es $x_i',y_i' \in K_N$ mit $b_2x_i' = x_i$ und $b_2y_i' = y_i$. Aus $hb_2y_i' = hb_2x_{i+1}'$ folgt $gb_1y_i' = gb_1x_{i+1}'$ für alle $1\leq i<n$, weil c_2 injektiv ist. Mit $x = x_1'$ und $y = y_n'$ existiert nach Lemma 11 für alle $1\leq i\leq n$ ein Weg von b_1x_i' nach b_1y_i', so daß schließlich gb_1y von gb_1x erreichbar ist. Demnach liegt Z ganz in hB_2.

Wäre $h^{-1}Z$ kein Zyklus, dann gäbe es o.B.d.A. genau zwei verschiedene Knoten x und y in B_2 derart, daß y von x erreichbar ist und h x und y identifiziert. Nach II.12.1 gäbe es $x',y' \in K_N$ mit $b_2x' = x$ und $b_2y' = y$. Nach Lemma 11 gäbe es einen Weg von gb_1x' nach gb_1y', der, da p fast ist, mindestens eine Kante enthält. Aus $hb_2x' = hb_2y'$ folgt $gb_1x' = gb_1y'$, weil c_2 injektiv ist. Damit wäre G zyklisch.

Also ist $h^{-1}Z$ ein Zyklus.

13 <u>Lemma</u>

Sei P die Σ-Grammatik von 10 und $G \overset{*}{\underset{P}{\Longrightarrow}} H$ eine Ableitung mit G azyklisch. Dann gibt es für jeden Zyklus Z in H $i,j \in \omega$ so, daß Z in den Graphen



eingebettet ist.

Beweis:

Da alle Produktionen von P fast sind und den Bedingungen (1) und (2) genügen, folgt aus Lemma 12, daß jeder Zyklus in einem Graphen, der durch eine direkte Ableitung über P aus einem azyklischen Graphen hervorgeht, mit dem Zyklus Z in der rechten Seite einer Produktion $p(t_{21,i,j})$ übereinstimmt. Andererseits wird ein solcher Zyklus durch eine direkte Ableitung über $p \in P$ nicht mehr verändert. Da p Σ-Produktion ist, müßte sonst einer der beiden Knoten von Z, x_t oder x_5, mit der Wurzel von p (in B_1) zusammenfallen. Das kann auf x_5 nicht zutreffen, weil die Wurzel von p nicht mit ATTR-X_i markiert ist. x_t kann wegen der Markierung von x_2 nicht Wurzel von p (in B_1) sein.

14 Lemma

Sei P die Σ-Grammatik von 10 und $\mathcal{G}$ die Menge aller aus azyklischen Graphen über P abgeleiteten Σ-Graphen. P ist schwach unabhängig bzgl. $\mathcal{G}$.

Beweis:

Sei $p \in P$, $G \in \mathcal{G}$ und $G \overset{p}{\Longrightarrow} H$ eine direkte Ableitung. Gäbe es $x \in K-\{x_p\}$ mit $gb_1x = gb_1x_p$, dann würde gb_1x_p wegen (1) zu einem Zyklus in G gehören. Nach Lemma 13 wären gb_1x_p mit EVAL oder ATTR-X_i und im ersten Fall der "linke Nachfolger" von gb_1x_p mit LAMBDA-X_j markiert. Das ist wegen $p \in P$ jedoch nicht möglich. Daraus folgt $gb_1x_p \neq gb_1x$ für alle $x \in K-\{x_p\}$ (IV.9.1).

Ferner ist zu zeigen, daß für alle $(p,\overline{p}) \in P\times P$ ein Schnittpunkt $\ddot{u}[p,\overline{p}]$ existiert mit

$$G \in \mathcal{G},\ G \overset{p}{\Longrightarrow} H,\ G \overset{\overline{p}}{\Longrightarrow} \overline{H},\ e \in B_{1,E},\ \overline{e} \in \overline{B}_{1,E}-\overline{b}_1\overline{K},\ ge = \overline{ge}$$
$$\Longrightarrow\ gs_1e = \ddot{u}[p,\overline{p}],\ gb_1x_p \neq \overline{gx} \text{ für alle } \overline{x} \in \overline{B}_1. \quad (3)$$

Die Existenz von $\ddot{u}[p,\overline{p}]$ brauchen wir natürlich nur für solche Paare $(p,\overline{p}) \varepsilon P{\times}P$ zu zeigen, für die es $G \varepsilon \mathcal{G}$ mit der Prämisse von (3) gibt. In der folgenden Tabelle sind alle diese Paare zusammen mit ihren Schnittpunkten aufgeführt. Wir verwenden im folgenden vereinfachte Bezeichnungen für die Produktionen von P, z.B. entspricht p_1 jedem Element der Menge $\{p(t_{1,\sigma}) \mid \sigma \varepsilon \Sigma\}$.

$(p,\overline{p})$	$\ddot{u}[p,\overline{p}]$
$\{(p_1,p_j) \mid 1 \leqslant j \leqslant 21$, wenn $j \leqslant 16$, dann j ungerade$\}$	x_2
$\{(p_2,p_j) \mid 1 \leqslant j \leqslant 16$ und j gerade$\}$	x_2
$\{(p_i,p_j) \mid i=3,13,15$, $1 \leqslant j \leqslant 21$, wenn $j \leqslant 16$, dann j ungerade$\}$	x_3
$\{(p_i,p_j) \mid i=4,14,16$, $1 \leqslant j \leqslant 16$ und j gerade$\}$	x_3
$\{(p_{17},p_j) \mid 1 \leqslant j \leqslant 21$, wenn $j \leqslant 16$, dann j ungerade$\}$	z_k
$\{(p_{19},p_j) \mid 1 \leqslant j \leqslant 16$ und j gerade$\}$	x_1

Da $\ddot{u}[p,\overline{p}]$ in $V(p,\overline{p})$ (vgl. IV.9) mit $\overline{b}_1 x_{\overline{p}}$ zusammenfällt, ist jeder Knoten $\overline{x} \varepsilon \overline{B}_1$ in $V(p,\overline{p})$ von $b_1 x_p$ erreichbar, aber nicht mit $b_1 x_p$ identifiziert. Deshalb impliziert $\overline{gx} = gb_1 x_p$ genau wie $gb_1 x = gb_1 x_p$ einen Widerspruch (s.o.). Also ist (3) erfüllt.

15 <u>Lemma</u>

Sei P die Σ-Grammatik von 10,
$P' = P - \{p_{21}\} \cup \{$ p collapse- oder delete-Regel $\mid m_1 x_p \varepsilon \Sigma\}$
und $\mathcal{G}$ die Menge aller aus azyklischen Graphen über P' abgeleiteten Σ-Graphen. P' ist schwach Church-Rosser bzgl. $\mathcal{G}$.

Beweisskizze:

Wie man sich leicht klarmacht, können collapse- und delete-Regeln keine Zyklen erzeugen. Damit gilt nach Lemma 14 die schwache Unabhängigkeit von P bzgl. $\mathcal{G}$. Aus der Tabelle in Lemma 14 ersieht man, daß P sogar stark unabhängig ist. Die Menge der P zugrundeliegenden Transformationsregeln ist abgeschlossen. p_1-p_4 und p_{13}-p_{18} sind zeigerumsetzend, p_5-p_{12} sind unzusammenhängend. Daraus folgt nach Satz IV.18, daß $P'' = P-\{p_{19},p_{20},p_{21}\} \cup \{p \text{ collapse-Regel} \mid m_1 x_p \varepsilon \Sigma\}$ schwach Church-Rosser bzgl. $\mathcal{G}$ ist.

Sei $Q = \{p_1,p_3,p_{13},p_{15},p_{17}\} \cup \{p \text{ collapse-Regel} \mid m_1 x_p \varepsilon \Sigma\}$ und $Q' = \{p_j \mid 1 \leqslant j \leqslant 16 \text{ und } j \text{ gerade}\}$. Nach Satz IV.10 und der Tabelle in Lemma 14 bleibt für die schwache CR-Eigenschaft von $P'' \cup \{p_{19},p_{20}\}$ zu zeigen, daß für alle

$$(p,\overline{p}) \;\varepsilon\; Q \times \{p_{19},p_{20}\} \cup \{p_{19}\} \times Q'$$

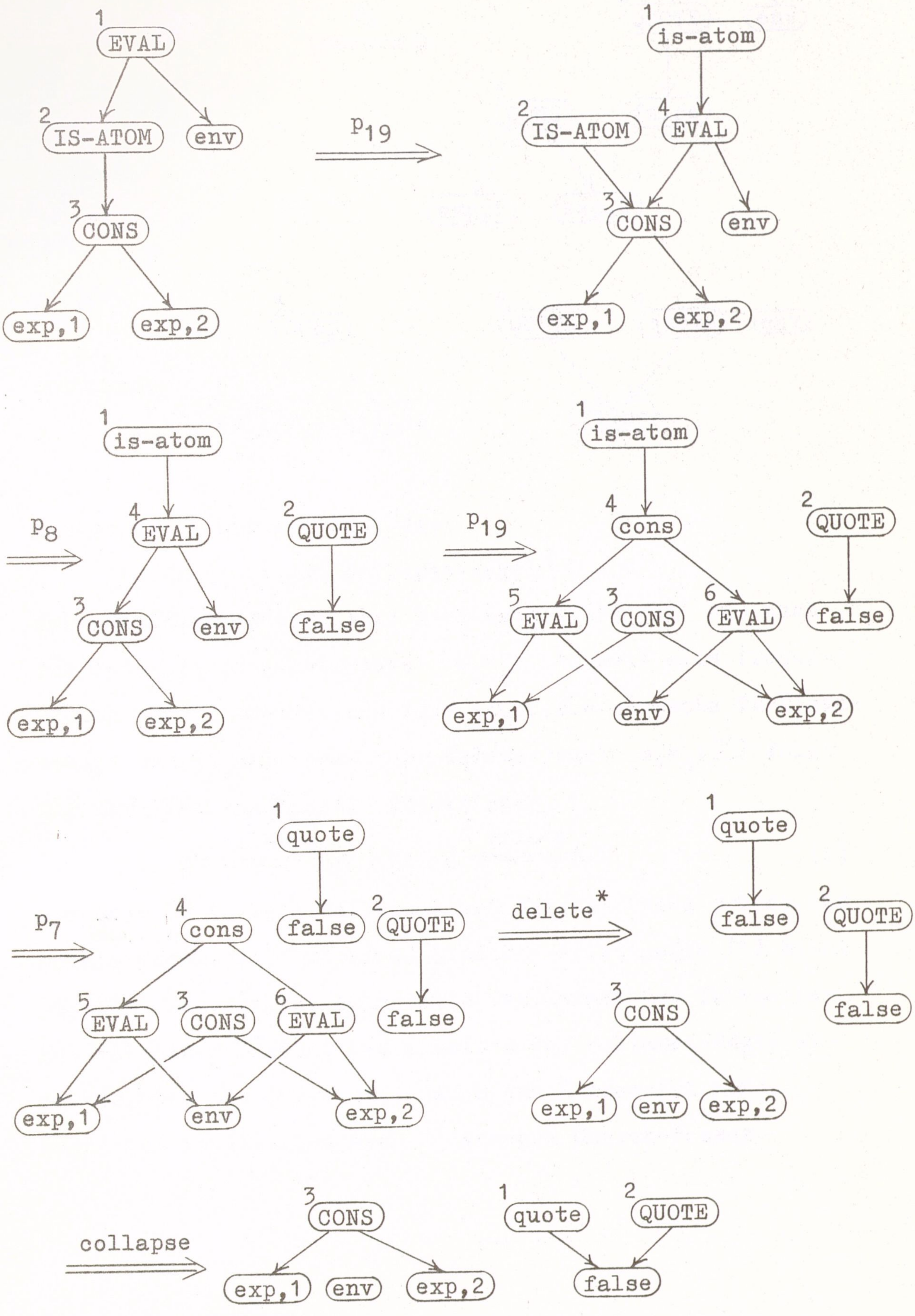
zwei in $V(p,\overline{p})$ beginnende konvergierende Ableitungssequenzen existieren, welche die Voraussetzungen von Satz IV.10 erfüllen.

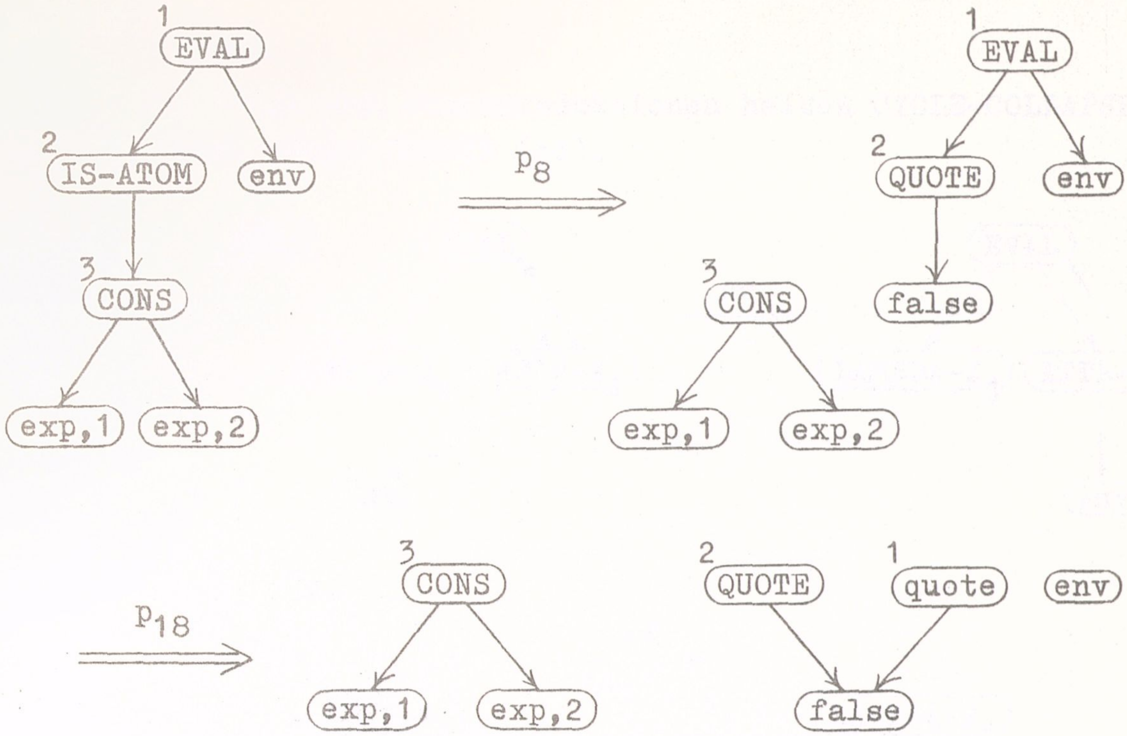
1) Sei $(p,\overline{p}) \;\varepsilon\; Q \times \{p_{19},p_{20}\}$. Da p zeigerumsetzend oder eine collapse-Regel ist, erhält man für $(p,\overline{p})$ zwei zu IV.18.4 bzw. zu IV.18.6 analoge Ableitungssequenzen.

2) Sei $\overline{p} \;\varepsilon\; Q'$. Dann gibt es $1 \leqslant j \leqslant 16$, j gerade, mit $p_j = \overline{p}$. Die beiden Ableitungssequenzen für (p_{19},p_j) sind in der folgenden Tabelle durch ihre Produktionsfolgen wiedergegeben. (p^2 steht für p,p)

j	Produktionsfolgen der beiden in $V(p_{19},p_j)$ beginnenden konvergierenden Ableitungssequenzen
2,4	$(p_{19},p_j,p_{19},p_{j-1},p_{19},\text{delete}^*)$ und (p_j,p_{19})
6	$(p_{19},p_j,p_{18},p_{j-1},\text{delete},\text{collapse})$ und (p_j,p_{18})
8	$(p_{19},p_j,p_{19},p_{j-1},\text{delete}^*,\text{collapse})$ und (p_j,p_{18})
10,12	$(p_{19},p_j,p_{18}^2,p_{j-1},\text{delete}^*,\text{collapse})$ und (p_j,p_{18})
14,16	$(p_{19},p_j,p_{18},p_{19},p_{j-1},\text{delete}^*)$ und (p_j,p_{19})

Exemplarisch für diese Fälle geben wir die Ableitungssequenzen des Paares (p_{19}, p_8) im einzelnen an:





Bisher haben wir gezeigt, daß

$$P-\{p_{21}\} \cup \{p \text{ collapse-Regel} \mid m_1 x_p \varepsilon \Sigma\}$$

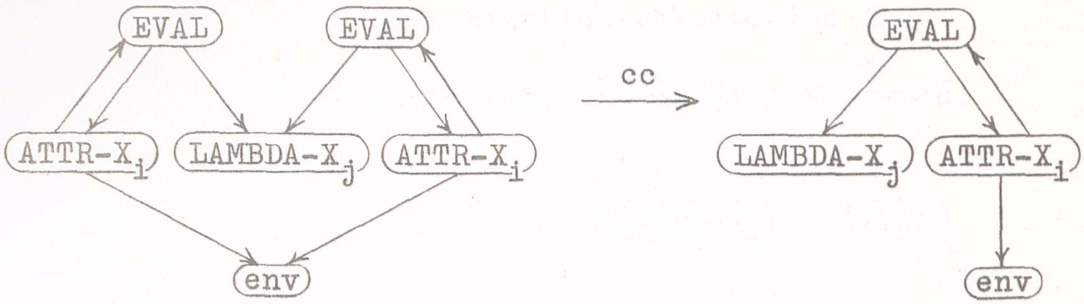
schwach Church-Rosser ist. Nach Satz IV.29 gilt das auch für $P-\{p_{21}\} \cup \{p \text{ delete-Regel} \mid m_1 x_p \varepsilon \Sigma\}$, weil alle Produktionen von P treelike und fast sind. Die schwache CR-Eigenschaft von $\{p \text{ collapse- oder delete-Regel} \mid m_1 x_p \varepsilon \Sigma\}$ folgt für parallel unabhängige Ableitungen

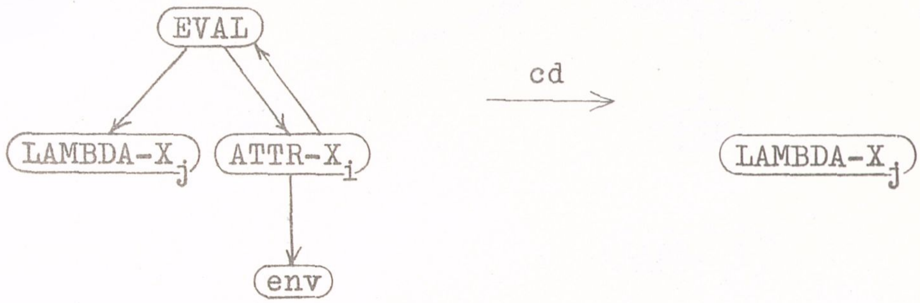
$$G \xRightarrow{\text{collapse}} H \quad \text{und} \quad G \xRightarrow{\text{delete}} \bar{\bar{H}} \tag{1}$$

aus Satz IV.4, weil collapse- und delete-Regeln proper und markierungskonsistent sind. Sind die Ableitungen (1) nicht parallel unabhängig, dann fällt der Ansatz der delete-Regel mit einem Büschel des Ansatzes der collapse-Regel zusammen, was zur Folge hat, daß H und $\bar{\bar{H}}$ isomorph sind.
Damit ist schließlich auch P' schwach Church-Rosser.

16 Definition

Die folgenden Graphproduktionen heißen CYCLE-COLLAPSE (cc) bzw. CYCLE-DELETE (cd).





Die linke Seite von cc und die rechte Seite von cd entsprechen den jeweiligen Klebegraphen.

17 Satz

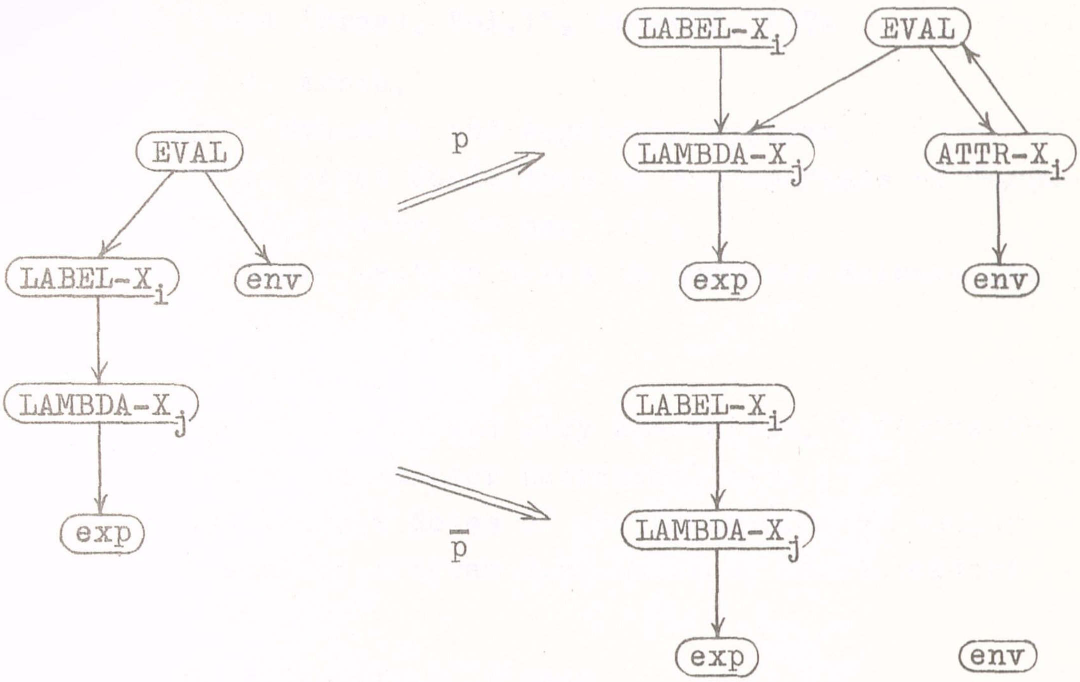
Sei P die Σ-Grammatik von 10,

$P' = P \cup \{\, p$ collapse- oder delete-Regel $\mid m_1 x_p \varepsilon \Sigma \} \cup \{cc, cd\}$ und $\mathcal{G}$ die Menge aller aus azyklischen Graphen über P' abgeleiteten Σ-Graphen. P' ist schwach Church-Rosser bzgl. $\mathcal{G}$.

Beweisskizze:

Sei Q wie in Lemma 15, p ε Q und $\overline{p} = p_{21}$. Dann ist p zeigerumsetzend oder eine collapse-Regel, und man erhält für $(p,\overline{p})$ zwei in $V(p,\overline{p})$ beginnende konvergierende Ableitungssequenzen analog zu IV.18.4 bzw. IV.18.6. Im letzten Fall ist collapse* durch cc zu ersetzen. Damit ist nach Satz IV.10 und Lemma 14/15 $P \cup \{p$ collapse-Regel $\mid m_1 x_p \varepsilon \Sigma\} \cup \{cc\}$ schwach Church-Rosser.

Sei $p = p_{21}$, $\bar{p}$ delete-Regel, $G \xRightarrow{p} H$ und $G \xRightarrow{\bar{p}} \bar{H}$.
Ist $\bar{g}x_{\bar{p}} \notin gB_1$, dann folgt die schwache CR-Eigenschaft von $\{p,\bar{p}\}$ wie im Beweis von Satz IV.31,Fall 1. Bei $\bar{g}x_{\bar{p}} \in gB_1$ lassen sich die folgenden direkten Ableitungen in $G \xRightarrow{p} H$ bzw. $G \xRightarrow{\bar{p}} \bar{H}$ einbetten, und es gilt $H \xRightarrow{cd} \bar{H}$.



$\{cc, cd\}$ ist aus den gleichen Gründen wie

$$\{\, p \text{ collapse- oder delete-Regel} \mid m_1 x_p \in \Sigma \,\}$$

schwach Church-Rosser (vgl. Lemma 15).
Folglich gilt mit Lemma 15 die schwache CR-Eigenschaft für P'.

REFERENZEN

/Ada 74/ J. Adamek,
Free Algebras and Automata Realizations in the Language of Categories,
Commentationes Mathematicae Universitatis Carolinae (Prag), Vol.15, 589-602, 1974

/Arb 77/ M. A. Arbib,
Free Dynamics and Algebraic Systems,
Proc. First Conference on Fundamentals of Computation Theory, Poznan 1977,
Springer Lecture Notes in Computer Science 56, 212-227

/Bau 78/ F. L. Bauer,
Detailization and Lazy Evaluation, Infinite Objects and Pointer Representation,
in: Lecture Notes of the International Summer School on Program Construction, Marktoberdorf 1978

/BL 70/ G. Birkhoff, D. Lipson,
Heterogeneous Algebras,
J. Combinatorial Theory 8 (1970), 115-133

/CM 72/ J. M. Cadiou, Z. Manna,
Recursive Definitions of Partial Functions and their Computations,
SIGPLAN Notices, Vol.7, No.1, 58-65, 1972

/ED 75/ E. Denert, H. Ehrig,
Mehrdimensionale Sprachen,
Vorlesungsskript, Technische Universität Berlin FB 20, 1975

/Ehr 78/ H. Ehrig,
Introduction to the Algebraic Theory of Graph Grammars,
Forschungsbericht Technische Universität Berlin FB 20 Nr. 78- , 1978

/EK 75/ H. Ehrig, H.-J. Kreowski,
Church-Rosser-Theorems leading to Parallel and Canonical Derivations for Graph Grammars,
Forschungsbericht Technische Universität Berlin FB 20 Nr. 75-27, 1975

/EK 75 a/ H. Ehrig, H.-J. Kreowski,
Categorical Approach to Graphic Systems and Graph Grammars,
Conference Report Algebraic System Theory, Udine 1975,
Springer Lecture Notes in Economics and Mathematical Systems 131, 323-351

/EK 76/ H. Ehrig, H.-J. Kreowski,
Contributions to the Algebraic Theory of Graph Grammars,
Forschungsbericht Technische Universität Berlin FB 20 Nr. 76-22, 1976

/Elg 73/ C. C. Elgot,
Monadic Computation and Iterative Algebraic Theories,
IBM Research Report RC 4564, 1973

/ER 76/ H. Ehrig, B. K. Rosen,
Commutativity of Independent Transformations on Complex Objects,
IBM Research Report RC 6251, 1976

/ER 77/ H. Ehrig, B. K. Rosen,
The Mathematics of Record Handling,
Proc. Fourth International Colloquium on Automata, Languages and Programming, Turku 1977,
Springer Lecture Notes in Computer Science 52, 206-220

/GHM 76/ J. V. Guttag, E. Horowitz, D. R. Musser,
Abstract Data Types and Software Validation,
University of Southern California Information Sciences Institute RR-76-48, 1976

/GTW 76/ J. A. Goguen, J. W. Thatcher, E. G. Wagner,
An Initial Algebra Approach to the Specification, Correctness and Implementation of Abstract Data Types,
IBM Research Report RC 6487, 1976

/GTWW 77/ J. A. Goguen, J. W. Thatcher, E. G. Wagner, J. B. Wright,
Initial Algebra Semantics and Continuous Algebras,
J. ACM, Vol.24, No.1, 68-95, 1977

/HM 76/ P. Henderson, J. H. Morris,
A Lazy Evaluator,
Proc. Third ACM Symposium on Principles of Programming Languages (1976), 95-103

/Kle 36/ S. C. Kleene,
λ-Definability and Recursiveness,
Duke Mathematics Journal 2 (1936), 340-353

/LS 77/ D. J. Lehmann, M. B. Smyth,
Data Types,
University of Warwick, Dept. of Computer Science, Theory of Computation Report No. 19, 1977

/Niv 74/ M. Nivat,
On the Interpretation of Recursive Program Schemes,
Universität Saarbrücken, Technical Report A74/09, 1974

/Pad 77/ P. Padawitz,
Church-Rosser-Properties for a Class of Graph Productions with Applications to Hyper-Pure LISP,
Technische Universität Berlin FB 20,
Interner Forschungsbericht, 1977

/Ros 73/ B. K. Rosen,
Tree-Manipulating Systems and Church-Rosser Theorems,
J. ACM, Vol.20, No.1, 160-187, 1973

/Scott 70/ D. Scott,
Outline of a Mathematical Theory of Computation,
Oxford University Computing Laboratory,
Technical Monograph PRG-2, 1970

/Scott 71/ D. Scott,
The Lattice of Flow Diagrams,
in: Semantics of Algorithmic Languages (ed. by E. Engeler), Springer Lecture Notes in Mathematics 188, 311-366, 1971

/Scott 72/ D. Scott,
Continuous Lattices,
in: Toposes, Algebraic Geometry and Logic (ed. by F.W. Lawvere), Springer Lecture Notes in Mathematics 274, 97-136, 1972

/Scott 74/ D. Scott,
Data Types as Lattices,
Proc. International Summer Institute and Logic Colloquium, Kiel 1974,
Springer Lecture Notes in Mathematics 499,
und SIAM Journal on Computing 5 (1976), 522-587

/Tar 55/ A. Tarski,
A Lattice-Theoretical Fixpoint Theorem and its Applications,
Pacific J. Math. 5 (1955), 285-309

/Vui 74/ J. Vuillemin,
Correct and Optimal Implementations of Recursion in a Simple Programming Language,
J. Comp. Syst. Sci., Vol.9, No.3, 332-354, 1974

/Wand 76/ M. Wand,
First-Order Identities as a Defining Language,
Indiana University, Computer Science Dept.,
Technical Report No. 29, 1976

INDEX DER DEFINITIONEN